THE

COMSTAR SYSTEM 4

REFERENCE MANUAL

COMSTAR CORPORATION
7413 Washington Avenue South
Edina, MN 55435

Publication No.
0973-7020

PREFACE

When computers were first introduced in the 1950's people everywhere believed that the computer had to be large and expensive. But in the mid 1960's minicomputers came into the market and disproved that a computer had to be a million dollar giant. Now is the time for everyone to realize that a microcomputer is able to do most everything that a mini-computer can for less cost. Besides the cost, microcomputers are flexible and are able to make changes as requirements change.

The reader is probably well aware of the fact that any computer can be divided into hardware and software. Hardware is the physical equipment while software is the communicating interface between the man and the machine.

COMSTAR Corporation is dedicated to develop and design industrial control systems using the microcomputer concept. Almost from the conception of a monolithic microcomputer, COMSTAR has been developing the methods of both hardware and software to apply the microprocessor to electrically noisy and environmentally rugged applications.

The market areas COMSTAR builds its equipment for is:

A. Machine Control and Testing

The area is defined by any system which uses closed loop control for mechanical motion.

B. Remote Monitoring and Control

This area uses modern data communication techniques to control systems or process over long distances.

C. Data Entry and Processing

This area means the source data is entered by manual means and processed into some permanent media form.

D. Material Handling

In this market area product is moved or packaged to be moved for storage and eventual distribution. The microcomputer controls the conveyor, sortation, packaging or palletizing and stacker crane equipment.

E. Industrial Computer Control

In this application, the microcomputer is used as a controller to control automatic assembly machines and on line testing equipment.

F. Traffic Control

The microcomputer is particularly effective as a controller in this area. The microcomputer can control intersections and monitor traffic or do time sequencing for dial type controllers.

This manual explains in detail the hardware and the software of the COMSTAR System 4 microcomputer. Hardware design of a digital computer depends upon individual company's useage of binary logic. Binary logic is mathematics dealing with the relationships among conditions based on a base of two. Those who are not familiar with binary logic should refer to Appendix A before attempting to really understand this computer's hardware. Appendix B provides the definitions of most of the technical terms. The reader is referred to Appendix B whenever he finds a new term and does not know the definition of it.

This manual will be updated time to time and suggestions and comments by readers are encouraged.

ABBREVIATIONS USED IN THIS MANUAL

A.C.	Alternating Current
ACC	ACCumulator
A/D	Analog to Digital
ASR	Automatic Send/Receive
BIT	Binary DigIT
BYTE	A Set of 4 Bits
CHIP	Integrated Circuit
CPU	Central Processor Unit
C4	Comstar 4
CS4	Comstar System 4
D/A	Digital to Analog
D.C.	Direct Current
D/C BUS	Data Control Bus
DTL	Diode-Transistor Logic
HTL	High-Threshold Logic
IC	Integrated Circuit
I/O	Input/Output
IR	Index Register
LED	Light-Emitting-Diode
LSI	Large Scale Integration
MOS	Metal Oxide Semi conductor
MUX	MUltipleXer
OEM	Original Equipment Manufacturer
OP CODE	OPeration Code
OPA	OPerand Address
OPR	OPerand Register

PAGE	Equivalent to 256 Instructions
PC BOARD	Printed Circuit Board
PCC	Process Control Compiler
PCL	Process Control Language
PORT	Input/Output (4 bit wide)
PROM	Programmable Read Only Memory
RAM	Random Access Memory
RTL	Resistor-Transistor Logic
T^2L	Transistor-Transistor Logic
TTL	Transistor-Transistor Logic
WORD	Equivalent to 4 bits (In CS4)

CHAPTER 1

1.0 General Discussion

The advances in electronics have changed both the technology and the economics of industrial digital control. The development of the minicomputer has vastly increased the scope of computer usage. In particular, the use of minicomputers in industrial control and signal processing applications cost generally between $5,000 and $15,000.

But the microcomputer represents still another electronic concept to the designer and users of industrial control equipment. Microcomputers are general-purpose digital circuits which can be programmed for a particular customer's requirement. It can be more easily changed or updated than a hard-wired logic system and can perform arithmetic as well as logic operation on applied signals. After the hardware system has been established it can be dedicated by software (in PROM's) for specific control applications.

Comstar Corporation is actively involved in producing micro-computer systems for industrial control applications. The Comstar System 4 can be divided into five basic units. They are --Comstar 4 microcomputer and interface modules; Machine Language PROM (Programmable Read Only Memory) programming equipment; PCL (Process Control Language) complier for PROM programming, the Program Analyzer to aid the testing of the programming, and the Comstar 4 Tester to allow the user to test the microcomputer, the memory and the interface modulars.

The Comstar 4 microcomputer is capable of the same arithmetic, control and computing functions of a minicomputer in as few small circuit

boards plus Power Supply at a fraction of the cost of a minicomputer system. Comstar 4 is as fully flexible and versatile as the best minicomputer, only the operating speed is slower. (Also less noise sensitive!)

The Comstar System 4 microcomputer is capable of performing most tasks currently done by minicomputers where speed capabilities of the minicomputer are not heavily taxed. This does not mean the CS4 is slow. For one application, the CS4 microcomputer is currently in use in a tape editing system reading and punching paper type at 110 characters per second and editing the data between characters. The Comstar 4 can execute up to 80,000 instructions per second.

1.1 Microcomputer: The heart of Comstar 4 is a single module central processor unit (CPU) which performs all control and data processing functions. Auxiliary to the CPU is the Programmable Read Only Memory (PROM) which stores microprograms and data tables, the Random Access Memroy (RAM) for data storage, and the Input/Output (I/O) modules which constitutes the I/O capacity of the system. The Comstar 4 communicates with circuits and devices outside the family through Input/Output "ports" provided on each I/O module or output "ports" on the small RAM module.

A Comstar 4 will usually consist of one CPU module, one or two PROM modules, one to four RAM modules, a Power Supply Regulator and up to 16 Input/Output modules.

A minimum system can be designed with just one CPU module, one PROM module, one RAM module, Power Supply, a Regulator and mouting rack. With these components, the buyer can build distributed,

dedicated, or personalized computers and utilize INFINITE COMBINATIONS OF MICROPROGRAMMING. The designer buys standard devices, and with microprogramming of the PROM, fulfills his own unique circuit requirements.

Chapter 2 discusses in detail the CPU, PROM, and RAM modules

1.2 Interface Modules

There are Digital, Power Switching, Analog, Communication, Peripheral Equipment and Special Interface Modules. The digital modules allow the user to have the C4 control digital input and output lines. Section 2.5.1 discusses the different types of digital modules. Power switching modules allow 12, 24, 48 or 120 volts of A.C. or D.C. Input and Output to be interfaced to the C4 microcomputer. Section 2.5.2 discusses all the types of power switching modules that are available.

Analog modules are 16 channel solid state analog multiplexer, 8 channel solid state differential multiplexer, 8 channel differential flying capacitor multiplexer, Sense Amplifier Module, analog to digital (A/D) convertors, Sample and Hold, and digital to analog (D/A) convertors. The multiplexers are capable of switching to one of 8 or 16 inputs at a time to an A/D convertor. 8 bit, 10 bit, 12 bit, 3 digit BCD and 4 digit BCD versions are available in the A/D and the D/A module. Details of these various versions is explained in section 2.5.3.

Section 2.5.4 discusses Data Communication modules. Serial asynchronous communication, serial synchronous and parallel data communication modules are available and depending on the data

transmission mode one of the modules can be selected.

Any auxiliary device through which data is input or output, to or from the microcomputer is called peripheral equipment. Paper tape readers, paper tape punches, magnetic tape cartridges or cassettes, card readers, disc memories, 6, 8, 16 and 32 digit displays, alpha-numeric printers and keyboards are all peripheral equipment. Section 2.5.5 discusses in detail about these devices.

The Real time clock module, Resettable clock module, the Pulse Accumulator, Internal Timer, Variable of Frequency, Time of Day and Quadrature Encoder modules come under Special Modules and information about these modules can be found in Section 2.5.7.

1.3 Machine Language Programming

The Programmable Read Only Memory (PROM) is capable of storing microprograms or data and is erasable by using ultraviolet light. To store microprograms, the PROM has to be programmed. Machine Language Programming is one of the two methods to program the PROM. In Chapter 3, beginning with Section 3.0, machine language programming is discussed. The machine language code is the most flexible and efficient method of programming the C4.

1.4 Process Control Language Programming

The ideal way to program any computer or controller is to do it in English language. Hence, Comstar has developed the Process Control Language (PCL) which is a straight forward and easy to use language. This language has similarities to FORTRAN and BASIC. All instructions in PCL are especially designed for the solution of problems in system operation.

The PCL instructions are loaded into the PROM memory with Comstar Process Control Compiler. The compiler converts PCL statements into C4 machine code.

1.5 Program Analyzer

Microprogramming is a technique of using a special set of instructions to execute a predetermined function at greater speed. The PROM is programmed with these special sets of microcode instructions. If the system doesn't perform according to the expectations, and if the hardware is sound, then there may be mistakes in the programming. Since the instructions in the PROM are executed at high speed and synchronously, there is no way of stopping at a particular address to check. This is where the Program Analyzer comes in.

The Program Analyzer displays the processor state (instructions, address, instructions, and data) either when the program reads a given address for the nth time ($1 \leq n \leq 16$), or when the nth external pulse is received. Procedure to operate the Program Analyzer and meaning of outputs on the Analyzer panel are explained in detail in Chapter 5.

1.6 System Tester

Comstar System 4 Tester can be arranged properly to test each module of the Comstar 4 microcomputer system. If there is a malfunction in one of the modules being tested, lights on the console will so indicate by not turning on and off in a defined unique sequence.

The basic Comstar System 4 Tester consists of a tested CPU, RAM, and PROM modules, Power Supply and Regulator module. Other modules and programs should be added for testing any other modules.

For example, a D.C. output module is needed to test a D.C. input module.

The basic Comstar System 4 Tester consists of a tested CPU, RAM, and PROM modules, Power Supply and Regulator module, other modules and programs should be added for testing any other modules. For example, a D.C output module is needed to test a D.C input module.

To operate a module, first the Tester must be equipped with the necessary modules, and programs. The PROM in the tester is programmed with the test programs and these programs can be executed by selecting the test number of the module on the thumbwheel switches. If the module under test is operational, the lights on the console will so indicate by turning off/on in accordance with the test program outputs. Refer to Chapter 6 for more details.

1.7.0 Application Summary

The CPU of the Comstar 4 has a powerful and versatile instruction set which allows the system to perform a wide variety of arithmetic, control and decision functions. The PROM, with microprograms, allows users the power of designing custom computers with standard components.

1.7.1 Machine Control and Testing

The main benefit of dedicating a Comstar 4 microcomputer to production machinery control is increased operational flexibility, since the entire system becomes programmable. With microcomputer control system, it is practical to build multifunction machines, since diverse tasks can be performed under the command of the controller.

An economic advantage is possible if one such machine can be used where several were previously necessary. Plus, programming may allow complex equipment to be modified for new production requirements, rather than fall into obsolescence. Examples are: 1) Automatic

armature coil winding machines controlled by the Comstar 4. The number of turns, turn counts and turn positons are fully programmable. Plus on line testing for part control can be simply implemented. 2) A bottling machine controlled by a Comstar 4 which provides instruction for the loading of bottles of different sizes and gives the command to initiate each step in the loading process.

A related characteristic of a machine with a dedicated microcomputer is the ability to be manually or automatically tuned for optimisim operation under a given set of conditions. An example is an automatic surface grinder whose stepping motors are tuned to give maximum performance by programming the stepping rate versus the load.

1.7.2 Romote Monitoring and Control

The microcomputer as a programmable remote station to be addressed from a central computer and be given operating setpoints strategies or can send data from a number sensors and then implement decision based on complex combination of conditions. In addition, distributed digital processing permits the central controller to receive refined status reports rather than raw data from remote stations. Since information is then exchanged only when established limits are exceeded, or when remote units are specifically interrogated, communication burdens are reduced. This can save money by reducing the complexity of the central computer programming and decrease the wiring or line cost by having each remote unit do more. Data can also be stored, timed and dated at a remote site by using a data storage peripheral unit such as magnetic tape cassettes. Finally, by relieving the central computer from routine data manipulation and control tasks, the machine can do more computations and better optimize the control of the total system.

1.7.3 Data Entry and Processing

In this application the microcomputer can often replace a dedicated minicomputer at a considerable savings in cost. The input data can be edited, sorted or verified using routines stored in the PROM and selected by a keyboard. An example of this is the editing and verification of the paper tape used in printing checks. The paper tapes have to be updated and checked before new checks can be printed. Other applications are the ability of the PROM memory for table storage. Thus, pricing, credit verification and inventory number checking can be stored in a non-volatile memory. An example of this is a point-of-pricing system which prices processed film and punches out paper tape on line from 8 stations for future invoicing. Finally, by doing the routine data collection and manipulations, the central computer can do supervisory functions such as scheduling and coordinating. This can yield a great economic benefit.

1.7.4 Material Handling

The dedication of microcomputers to operating machinery also can be considered a means of distributing intelligence through a system. Particular advantages can be derived from placing computing elements at interface between production or distribution equipment and the material or goods being processed or handled. For example, local signal processing makes it possible to establish local control loops. These can regulate specified variables under normal conditions without intervention from the central system. As an illustration a Comstar 4 micro-

computer controls a box palletizer which after it is told the pattern and number of tiers, builds pallet loads of boxes until it is told to change for a different size box.

Local control has long been accomplished using traditional elements such as relays, timers or counters. However, where specialized functions are to be performed, the cost associated with development of electromechanical controllers can exceed those required to implement the central processor. Thus, overall efforts can be reduced considerably by using standard general-purpose modulars, which can be programmed to meet the needs of individual systems.

Refer to Chapter 7 to understand how the microcomputer is specifically applied.

1.8 COMSTAR SYSTEM 4 FEATURES

* Central Processor on a single PC board with 45 instructions
* Decimal and binary arithmetic modes
* Up to 80,000 instructions can be executed in one second
* Nesting of subroutines up to three levels
* Erasable and reprogrammable PROMs as control memory
* Can add two 8 digit numbers in 850 microseconds
* A full line of interface modules are available
* Directly addresses 32 k bits of PROM
10.2 k bits of RAM
1024 I/O lines
* Memory expandable through bank switching
* Full line of PROM programming equipment available
* Program Analyzer available
* Full line module System Tester available
* Expandable power supply system
* Extensive use of metal-oxide semiconductor (MOS) circuitry

CHAPTER 2 Hardware Description of CS4 Microcomputer

2.0 Introduction

2.1 CPU Module

2.2 PROM Modules
- 2.2.1 Control PROM
- 2.2.2 Auxiliary PROM
- 2.2.3 PROM programmers

2.3 RAM Module
- 2.3.1 Version A
- 2.3.2 Version B
- 2.3.3 Large RAM (4K x 4)

2.4 Power Supply System

2.5 Interface Modules (As a class)
- 2.5.1 Digital
- 2.5.2 Power Switching
- 2.5.3 Analog
- 2.5.4 Communication
- 2.5.5 Peripheral Equipment
- 2.5.6 Specials

2.6 Interface Module Addressing (Intersil PROM)

2.7 Packaging
- 2.7.1 Card Guide
- 2.7.2 Panel Mounting (Single and Double)
- 2.7.3 Rack Mounting
- 2.7.4 Special Mounting

CHAPTER 2

COMSTAR 4 MICROCOMPUTER DESCRIPTION

2.0 "Do it yourself" is the idea behind COMSTAR 4 microcomputers. The basic microcomputer consists of one Central Processor Unit (CPU) module, one Programmable Read Only Memory (PROM) board with one PROM chip, one Random Access Memory (RAM) board with one RAM chip, a small Power Supply, one Regulator and a mounting rack. The basic System is expandable by adding more modules.

COMSTAR 4 SYSTEM OPERATION

TIMING

The COMSTAR 4 systems use a 4 bit micro-processor thus all data is communicated between the system elements in groups of 4 bits. The instruction cycle for the CPU requires eight, 4 bit time intervals as shown in Figure 2.00 . The eight time intervals accomplish program memory addressing, instruction retrieval, and instruction execution. The 12 bit address, required to address up to 4096 words of program memory, is sent from the CPU to the program memory in three time intervals defined as A1, A2, and A3. The eight bits of instruction for each word are sent from the addressed program memory to the CPU in two time intervals defined as M1 and M2. During the last three time intervals defined as X1, X2, and X3 the CPU executes the instruction.

Each time interval is generated by the operation of the two phase CPU clock circuit. The two phase clock accomplishes the operations within the MOS CPU, RAM, and CPU interface devices.

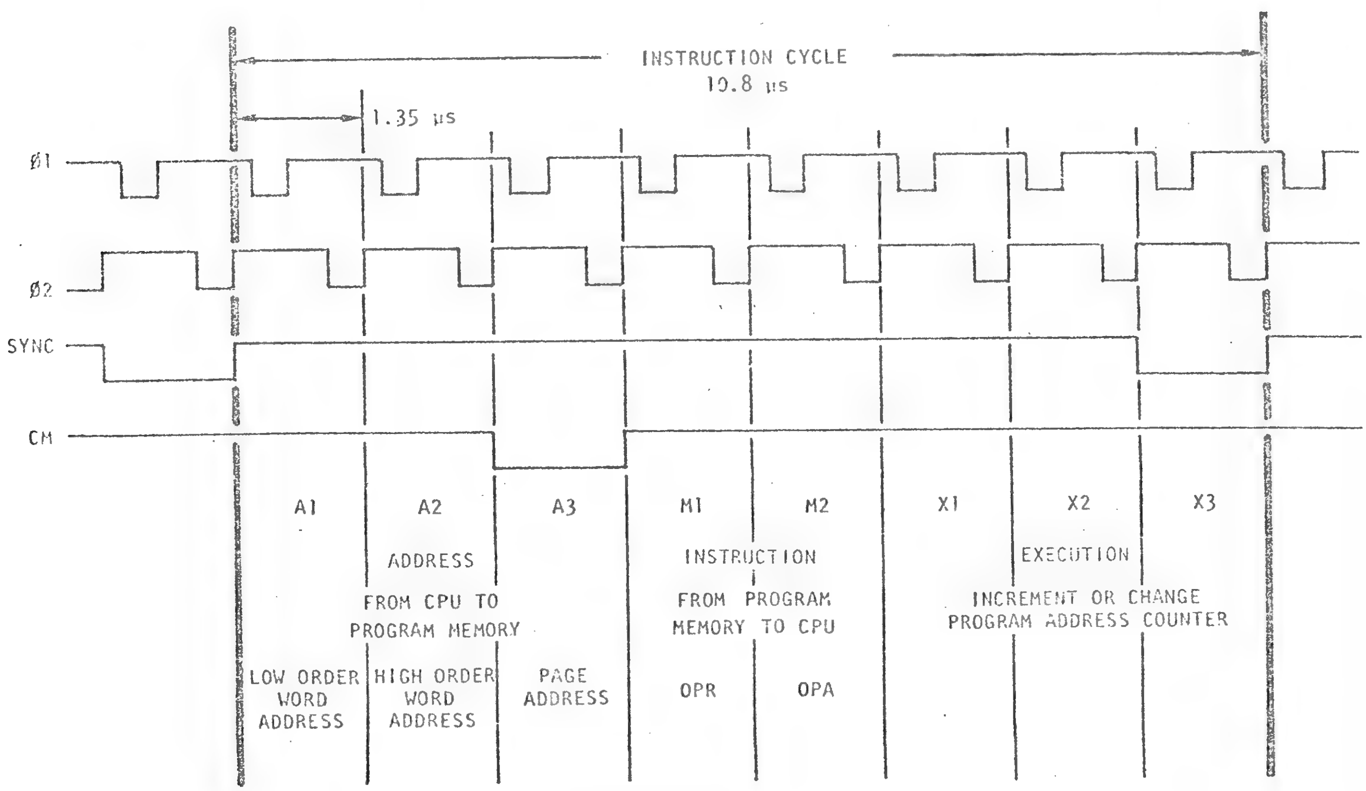


FIGURE 2.00

CPU Instruction Timing for Most Instructions

The sync pulse sent from the CPU keeps the RAM register and CPU interface devices in step with the CPU. The CM line signals the RAM registers and the CPU interface device to accept and decode chip select information on the CPU bus. CM always occurs at A3 time as this is the ROM program memory chip select (page) address. CM also occurs at X2 time as shown in Figure 2.01 during the SRC instructions for addressing RAM register devices and I/O ports, and at M2 time shown in Figure 2.02 during I/O and RAM register instructions for sending operand information to the RAM registers and the CPU interface circuits.

CM-RAM lines available from the CPU are used for bank switching of RAM register devices. If four or less RAM register devices are used on a system they may be tied to the CM line. When the CM-RAM lines are used and selected using the DCL instruction the timing is identical to the CM timing shown in Figures 2.00, 2.01 and 2.02.

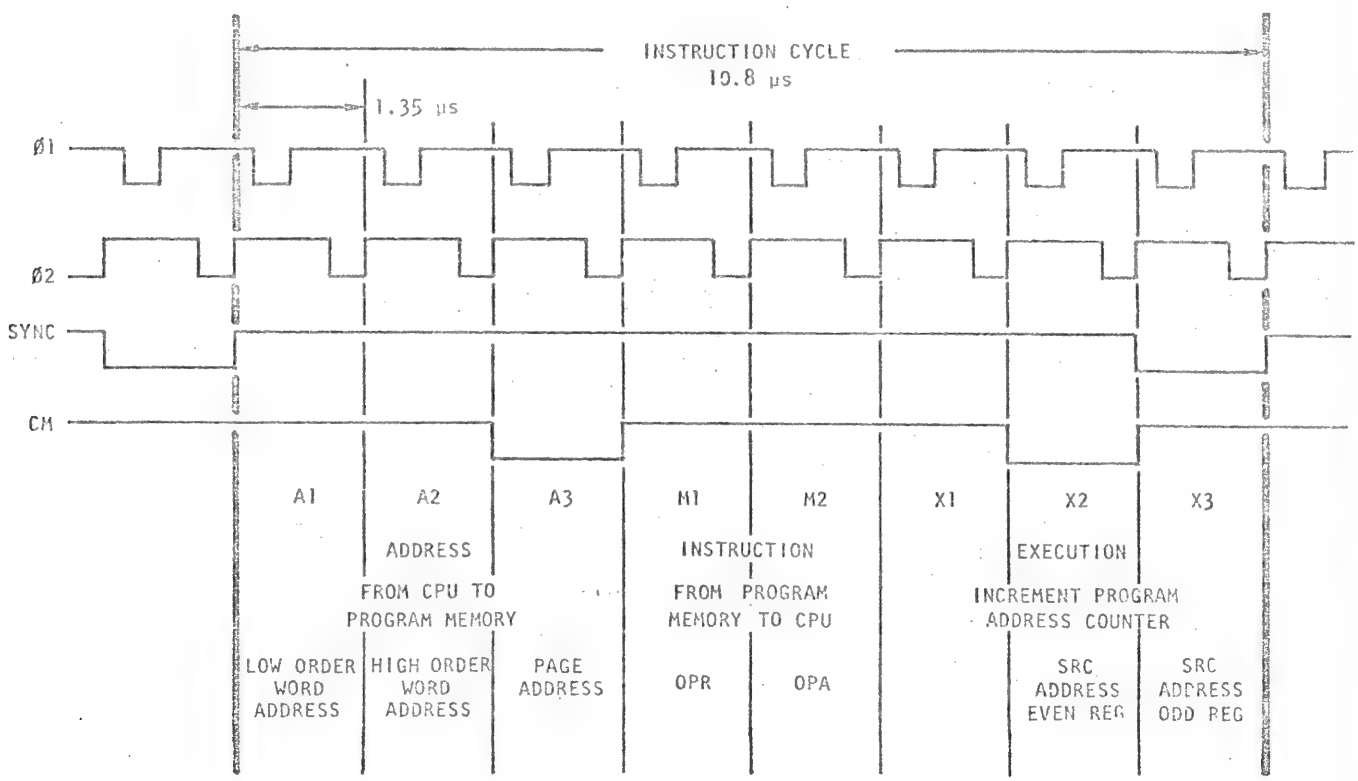


FIGURE 2.01

CPU Timing for SRC Instruction

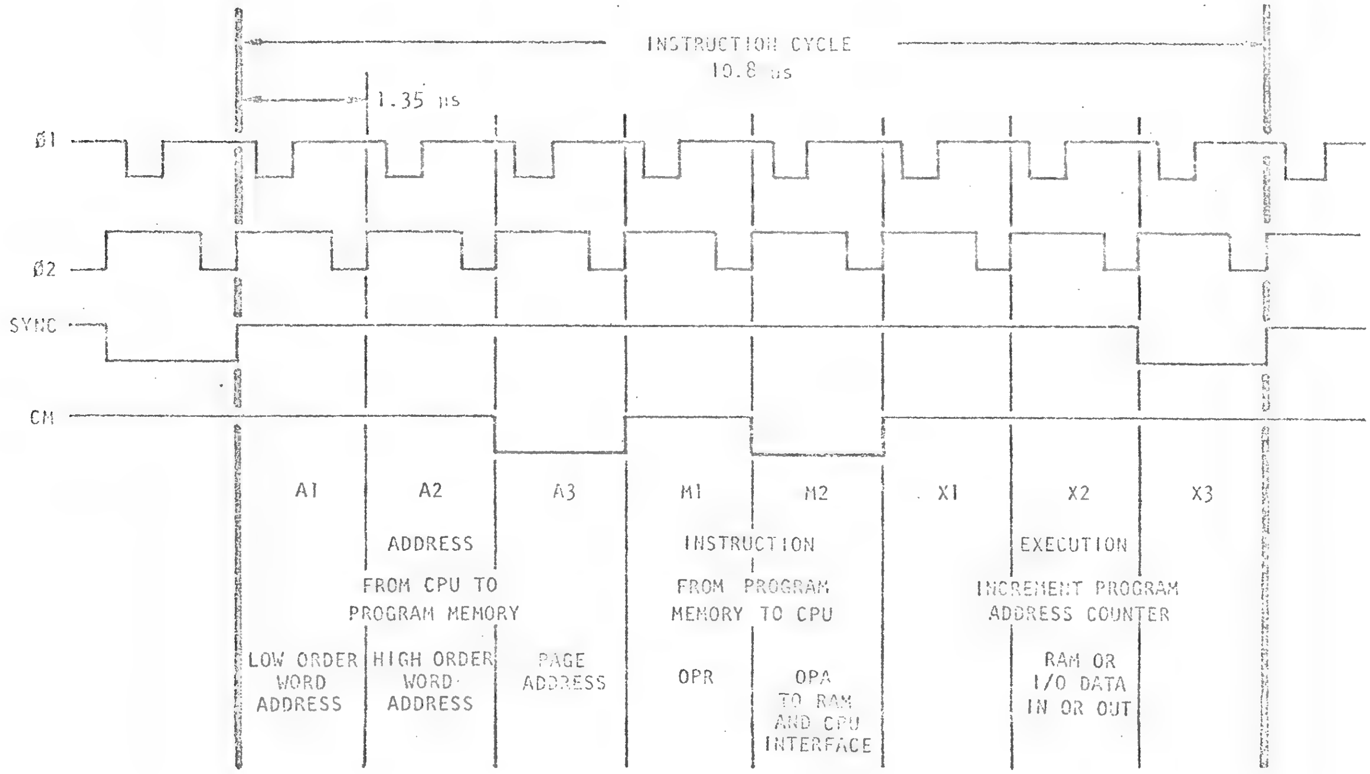


FIGURE 2.02
CPU Timing for I/O and RAM Register Instructions

2.1 Central Processor Unit Module

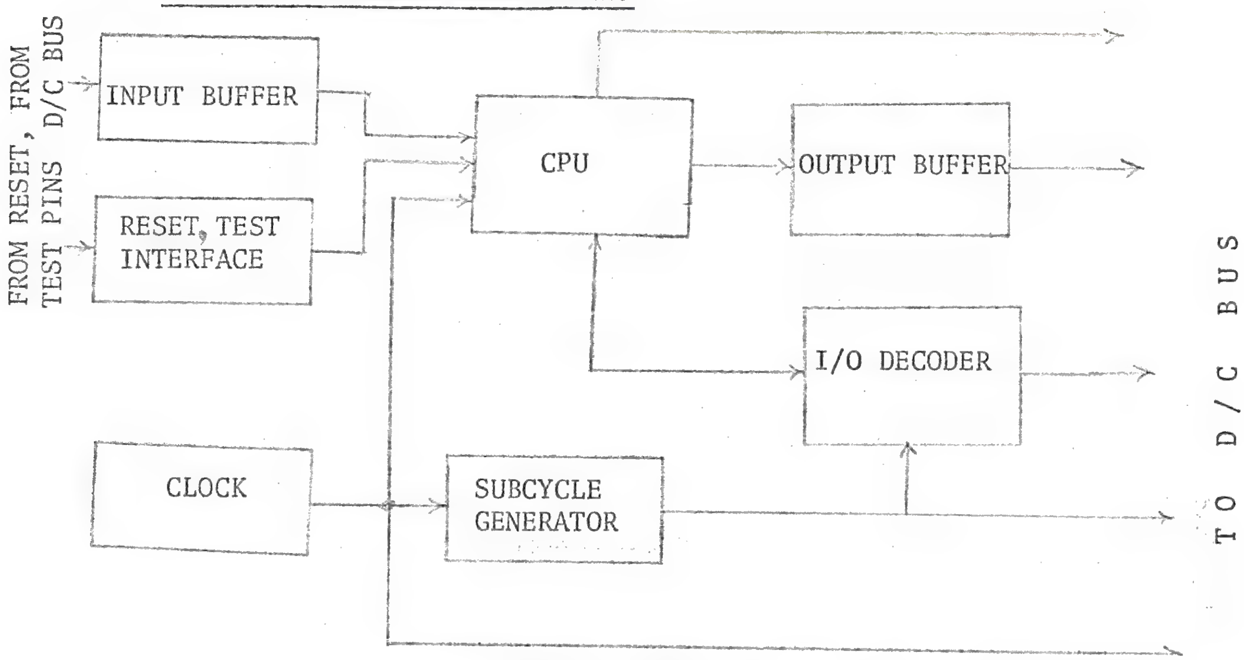


Figure 2.10

The CPU module contains Input Buffer, ~~Rest~~ Reset/Test interface, clock generator and subcycle generator, CPU unit and output Buffer/Decoder. Output of PROM module is transmitted to CPU unit via Input Buffer. RESET, TEST interface is capable of sensing signals. Clock generator generates two 12.0 microseconds non-overlapping clock phases -- $\emptyset_1$ and $\emptyset_2$. Sub cycle generator is capable of splitting the generated cycle.

The CPU unit contains the control unit and the arithmetic unit of a general purpose microprogrammable computer. The CPU communicates with the other members of the system through a data control bus, which is a 50 conductor flat cable system, and also receives power from this same cable. The CPU module contains five command control lines: four of which are used to control the RAM chips (each line can control up to 4 RAM chips for a total system capacity of 16 RAMS). One of which is used to control a bank of up to 16 PROM chips, (2 PROM modules).

DATA CONTROL BUS

LINE	SIGNAL	EXPLANATION
A1	GND	Ground
A2	-15V	
A3	SYNC	CPU Chip generates this signal for every 8 clock periods. i.e. SYNC Signal marks the beginning of each instruction cycle.
A4	Ø2L	Clock Phase 2
A5	Ø1L	Clock Phase 1
A6	TEST	HTL: Logic 1= GND T^2L: Logic 1= +5V To use T^2L TEST, ground at least one of the HTL test lines.
A7	MDB0	Memory data bits
A8	MDB1	
A9	MDB2	
A10	MDB3	
A11	RAM0	Memory Control outputs
A12	RAM1	
A13	RAM2	
A14	RAM3	
A15	REST	Logic 0, low= GND
A16	ROM0 (CM)	Memory Control output
A17	GND	Ground
A18	PDB0	Input data bits on T^2L bus. Logic 1, high= +5V
A19	PDB1	
A20	PDB2	
A21	PDB3	
A22	NDB0	Output data bits on T^2L bus. Logic 1, high= +5V
A23	NDB1	
A24	NDB2	
A25	NDB3	
A26	——	Not used
A27	——	Not used

A28	——	Not used
A29	——	Not used
A30	CIEN	Command Input Enable. Active during Logic= 0. RDR instruction activates a pulse on the line.
A31	CIRT	Command Input Reset. Active during Logic= 1. When RDR instruction is used, the instruction enables the computer to read the data (CIEN goes low) and then resets latched input.
A32	CØ2T	Clock phase 2 inverted, T^2L. This line selects a PROM Chip on the PROM board.
A33	CBM2	Clock bit M2, low during M2
A34	CBM1	Clock bit M1, low during M1
A35	CBA3	Clock bit A3, low during A3
A36	CBA2	Clock bit A2, low during A2
A37	CBA1	Clock bit A1, low during A1
A38	CØEN	Command Output Enable. Active during Logic= 0. WRR instruction activater 1μ sec. pulse on the line.
A39	CDB4	Bank select data lines, T^2L Logic 1= high
A40	CDB5	
A41	CDB6	
A42	CDB7	
A43	CBX3	Clock bit X3, low during X3
A44	CØAD	Command Output Address, T^2L. Input/Output address on the bus. Active during Logic 1= high. Active during SRC instruction
A45	GND	Ground
A46	+5V	
A47	GND	Ground
A48	+15V	
A49	GND	Ground
A50	——	Not used

2.2 PROGRAMMABLE READ ONLY MEMORY (PROM) MODULE

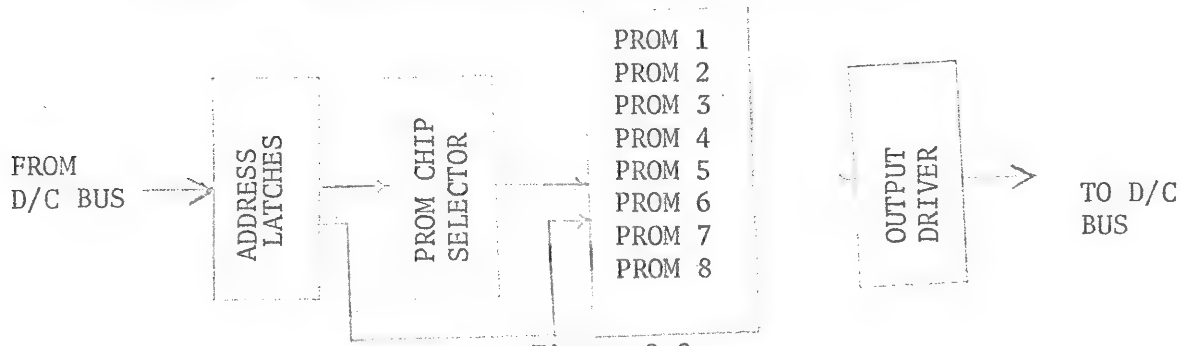


Figure 2.2

The PROM chip is a 2048 (256 x 8) bit electrically programmable and light erasable Read Only Memory, providing custom microprogramming capability for the CS4 microcomputer. Each chip is organized as 256 x 8 bit words which can be used for storing programs or data tables. On a PROM board up to 8 chips are allowed.

During A_1, A_2, A_3, of the instruction cycle, the PROM module accepts and decodes an address from the CPU. The chip select logic enables one of the eight chips on the card, and then during M_1 and M_2 of the instruction cycle, the demultiplexer puts the data from the PROM onto the data control bus.

2.2.1 Control PROM Module

The PROMS in the module contain machine language programs that control the operation of the Comstar 4 microcomputer. Each module can have up to 8 PROM chips (2K x 8 bits) and each system can have up to 2 modules (4K x 8 bits).

2.2.2 Auxiliary PROM Module

The auxiliary PROM chips are the same physically as the Control PROMs (256 x 8 bits). However, the auxiliary PROM module is interfaced as an I/O module which means that data stored on these PROMs can be accessed only as data. The PROMs therefore, cannot contain control programs to run the Comstar 4 microcomputer.

The PROM chips can be programmed with either one of the following machines:

2.2.3. PROM PROGRAMMING MACHINES

2.2.3.1 Manual Switches: The unit has 8 data switches and 8 address switches with associated L.E.D's indicating the on/off condition of each switch. The address to be programmed is set by the address switches, the data is set in by the data switches, the write mode is enabled and the data is entered into the PROM. This sequence is repeated for each address requiring data.

2.2.3.2 Keyboard: This unit allows input via keyboard, of the address and the data in hexadecimal value. The input is displayed on a 6 digit alphanumeric display which is part of the programming device. The left most two digits display the address, the middle display data, and the last 2 digits display the contents of the address shown in the left most display.

2.2.3.3 CPU Editor:: This has full editing capacity. That is, the device reads the PROM contents and stores it in RAM. The data can be edited, changed, and then written to PROM from RAM. The entire PROM chip is programmed serially from address 00 to FF. The Editor includes the Keyboard programmer and a Comstar 4 microcomputer. The computer is useable as a stand alone device.

2.2.3.4 Portable Editor: A portable unit with display and data entry keys to edit and program PROMs in machine language. Since it is a simple portable unit to use and the input is via a high speed paper tape reader - - the Portable Editor is preferred to the CPU Editor.

2.2.3.5 EAROM: A 2K, Electronically Alterable Read-Only Memory to use for debugging programs on the Comstar 4. The system includes an interface module, 16 key keyboard with five digit display and self contained Power Supply. The system allows trapping and stopping of programs at a pre-determined address.

2.2.3.6 Assembler

A typical assembler is accessed by a teletype machine with printer, paper tape reader, paper tape punch and keyboard. This unit is plugged into a Comstar or time sharing computer system and interfaced via a serial communication module. Magnetic tape is optional to replace the paper tape media.

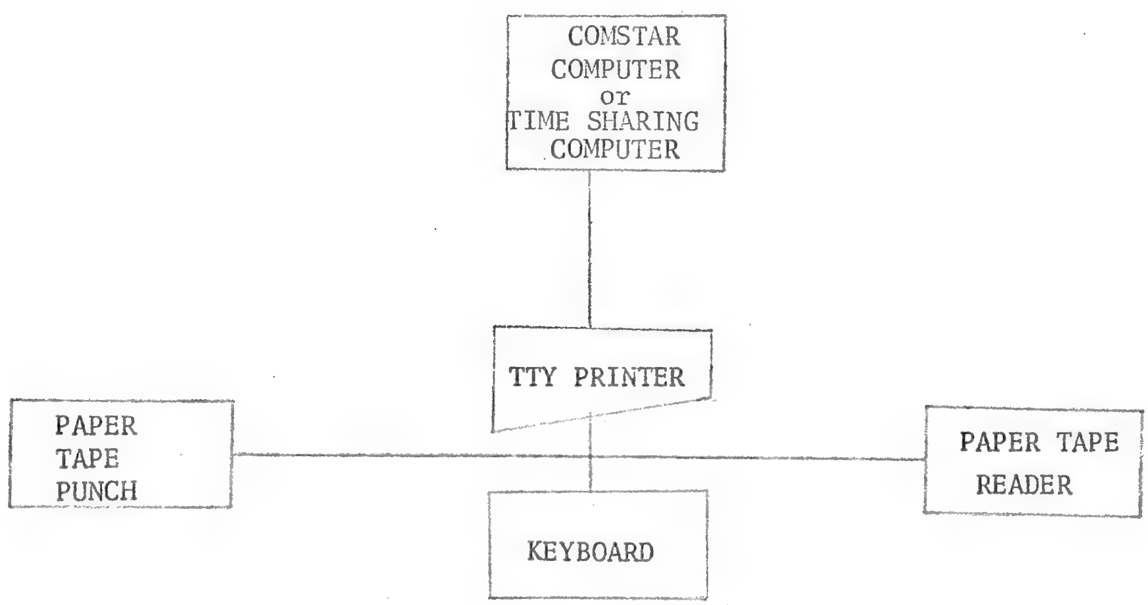


Figure 2.2.3.6

2.2.3.7 Process Control Compiler

A portable unit with display and data entry keys to edit and program PROMs in Process Control Language as well as in Machine Language. The Compiler can be obtained with a printer which can print the listing of a programmed PROM via Compiler.

2.3 RAM Module

Random Access Memory can be used to store data for future access by the control PROM. RAM memory is also known as Read-Write memory and as scratch memory. Comstar customers use three types of RAM modules, namely A Version, B Version and C Version.

2.3.1 A Version

This RAM module performs two functions. As a random access memory it has total capacity of 5.12K bits. As a vehicle of communications with display devices, each card has up to 32 output lines and associated control logic. Open collector outputs are standard, but totem pole outputs are optional on order.

2.3.2 B Version

The B version, RAM module, has a total capacity of 10.24K bits. This module cannot be mixed with the A version RAM module. Open collector outputs are standard, but totem pole outputs are optional on order.

2.3.3 C Version

This module can be configured as a 4K X 4 bit or 2K X 8 bit storage media and does not have output capabilities.

2.4 Power Supply and Regulator

Every C4 microcomputer system requires a Power supply and one or more regulators. At Comstar, three different types of power supplies are available — the small, medium and large.

The small power supply included in the basic system can supply up to 10 units of D.C. power. It is easy to compute power consumption for a particular system by using the Price Schedule sheet. Once the power consumption is determined, pick a desired power supply and number of regulators from the following table:

D.C. POWER IN UNITS	# OF REG.	TYPE OF POWER SUPPLY
0-10	1	Small
11-20	2	Medium
21-30	3	Large
31-40	4	Large
41-50	5	Large

If the total D.C. Power exceeds 50, then small, medium and large power supplies can be combined to achieve the required power.

2.5 INTERFACE MODULES

2.5.1 Digital Modules

MODULE	MODULE #	DESCRIPTION
32 Input,T^2L	9007-2100	This module allows 32 inputs (T^2L type input signals) to the C4 microcomputer.
32 Input,T^2L	9007-2101	As above, but photo-isolated inputs.
32 Input,HTL	9007-2200	As 9007-2100, but HTL type inputs.
32 Output,T^2L	9007-2300	This module provides 32 outputs (T^2L type output signals) from the C4 microcomputer.
32 Output,HTL	9007-2400	As above, but HTL type outputs.
16X16 I/O,T^2L	9007-2500	This module allows 16 (T^2L type input signals) to the C4 microcomputer and provides 16 outputs from the microcomputer.
16X16 I/O,HTL	9007-2600	As above, but HTL type inputs/outputs.
Mux Driver	9007-2700	The Multiplexer Driver module can interface 384 contact closures or 96 thumbwheel switches.
Multi-Purpose	9007-275X	This Non-Data bus module is for T^2L and HTL inputs to 9007-2700. Real time/Resettable clocks optional. Where X: 0, 1 or 2.
Diode Board	9007-2800	This module provides diode isolation between external contact devices and 9007-2700 module.

2.5.2 Power Switching Modules

MODULE	MODULE #	DESCRIPTION
A.C. Input	9007-3100	This module provides eight channels of optically isolated input to interface between A.C. industrial control voltages and the C4 microcomputer. 12,24,48 or 120 volts optional.
A.C. Output	9007-3200	This module provides eight channels of reed-relay isolated output to interface between A.C. industrial control voltages and the C4 microcomputer. 12,24,48 or 120 volts optional.
D.C. Input	9007-4100	This module provides eight channels of optically isolated input to interface between D.C. industrial control voltages and the C4 microcomputer. 12,24,48 or 120 volts optional.
D.C. Output	9007-4200	This module provides eight channels of reed-relay output to interface between D.C. industrial control voltages and the C4 microcomputer. 12,24,48 or 120 volts optional.

2.5.3 Analog Modules

MODULE	MODULE #	DESCRIPTION
Sense Amp	9007-5400	This module is ideal to condition from signals to a standard A/D module.
Flying Cap Multiplexer	9007-6000	This module has 8 channels of differential inputs providing 1000 volts isolation to the A/D from the input signals.
8 bit binary A/D	9007-61XX	8 Bit Analog to Digital module converts an analog voltage to 8 bit binary number. This module is available either with a 16 channel Analog multiplexer or a 8 channel differential multiplexer.
12 bit binary A/D	9007-62XX	As above, but 12 bits.
3 digit BCD A/D	9007-63XX	As above, but 3 Digit BCD
4 digit BCD A/D	9007-64XX	As above, but 4 Digit BCD
10 bit binary A/D	9007-65XX	As above, but 10 bit
2 digit BCD A/D	9007-66XX	As above, but 2 Digit BCD
Sample & Hold	9007-7000	4 or 8 channel Sample & Hold module can be used with a Digital to Analog (D/A) module.
8 bit binary D/A	9007-710X	8 Bit Digital to Analog (D/A) module converts digital information into analog voltages.
12 bit binary D/A	9007-720X	As above, but 12 bits
3 digit BCD D/A	9007-730X	As above, but 3 digit BCD
4 digit BCD D/A	9007-740X	As above, but 4 digit BCD
10 bit binary D/A	9007-750X	As above, but 10 bits
2 digit BCD D/A	9007-760X	As above, but 2 digit BCD

2.5.4 Communication Modules

MODULE	MODULE #	DESCRIPTION
Serial	9007-8100	This module allows C4 microcomputer to communicate with Peripheral devices. 110 to 2400 Baud rate available. This Serial asynchronous module can be interfaced with 103 type modem and teletype(TTY).
Serial	9007-8101	As above, but this module can be interfaced only with 202C type modem and auto answer circuits.
Parallel Comm	9007-8200	This module allows C4 microcomputer to communicate with Peripheral devices whose interface is parallel. The Parallel Data Communication module can be supplied positive or negative logic inputs or outputs in either T^2L or HTL
	9007-8300	This module(consists of 3 printed ckt boards) can be interfaced with 201 or 203 type modem .
Bose-Chaudhuri	9007-8301	The Bose-Chaudhuri cyclic redundancy check code module can generate 16 bits of cyclic redundancy check code (CRC code) when 16 natural binary bits are provided. It also can check for an error if 16 CRC bits are provided.

Communication Modules Continued -----------

MODULE	MODULE #	DESCRIPTION
Async Modem	9007-8350	The Asynchronous Modem(Modulate-Demodulate) module can be used to modulate or demodulate the digital information which is sent or received by the transmission lines. This module is rated at 1200 bits/second over the switched network and 1800 bits/second on private lines.
Sync Modem	9007-8351	As above, but synchronous operation.

2.5.5 Peripheral Equipment

Paper tape punch

A 35/70 character/second punch and interface module with read-after-punch verification. The unit is in use for 1 inch wide, 8 level tape.

Paper Tape Reader

A 200 character per second photoelectric reader and interface module. The reader comes standard set for 1 inch wide, 8 level tape but is optionally available to handle 5, 6, or 7 track tape or 6 track typesetting tape or Japanese telex.

Magnetic Card Reader

Interface module and reader for reading credit cards, with magnetic tape encoded information.

Magnetic Tape Cartridge

Interface module with magnetic tape cartridge unit for Comstar 4 cartridges. The unit has read-after-write verification with a buffered I/O channel.

Card Reader

Interface module and 150 card per minute reader (80 column) hopper/stacker capacity of 400 cards.

Disk Memory System

Interface module and cartridge disk system with a removable cartridge having a 500,000 bit storage capacity and an average access time of 16.7 microseconds.

Also available:

Standard cartridge plus a 125,000 bit fixed disk.

Standard cartridge plus a 250,000 bit fixed disk.

Disk cartridge for Disk Memory System - oxide coated, 1.5 mil mylar.

Display Driver

Drives the Burroughs SSD 1000-0030 (16 channel) or the SSDO 132-6030 (32 character) Alpha Numeric Display (does not supply high voltage for display).

Display Module

This is a six digit display with interface module and includes the decode capability and display driver. Output to the display is from the Comstar 4 RAM.

Keyboard Enclosure

This device is a 16 key keyboard with interface module, utilized for data entry. The keyboard has ten numeric keys and six additional keys for customer definition. (This device requires I/O Data Module.) Metal enclosure for the 16 keyboard.

Alpha-numeric Printer

Interface module and impact, dot matrix printer. Print capacity is 34 characters per line; print speed is 110 characters per second. Line speed is 1.5 to 4 lines/second.

Thumbwheel Switch Module

This module has 4 to 8 thumbwheels mounted on the top edge to provide BCD entry capability.

L.E.D. Display Module

16 L.E.D.'s mounted at right angles to the board provides a visual indication for checking logic level outputs from a program in PROM.

2.5.6 Specials

Pulse Accumulator

This module provides two 12-bit counters capable of accumulating pulses from two external sources for frequency measurement. Input pulses may be accumulated and then held for interrogation by the CPU on the basis of pre-selected (jumpered) time intervals ranging from 100 microseconds to 10 seconds.

Interval Timer Module

The interval time module accepts a 16-bit or 4 digit BCD number from the CPU and counts it down to zero in preset time increments. Two timers are provided in each module.

Quadrature Encoder

Interface for optical incremental encoder with HTL or T^2L quadrature outputs.

2.6 MODULE ADDRESSING

There are two levels of addressing. In the first level, one of sixteen banks (0 to F_{16}) will be selected and in the second level, one of sixteen addresses (0 to F_{16}) within the selected bank will be selected.

Totally there are 256 (16 banks x 16 addresses) uniqure addresses for device interface.

Each address location physically connects to four (eight in special cases) input/output lines (See figure 2.6.1, 2.6.2)

So, to address n^{th} location of bank Y,

First, select Bank Y

Second, select address N

Third, read/write to input/output four lines (line=bit) to information.

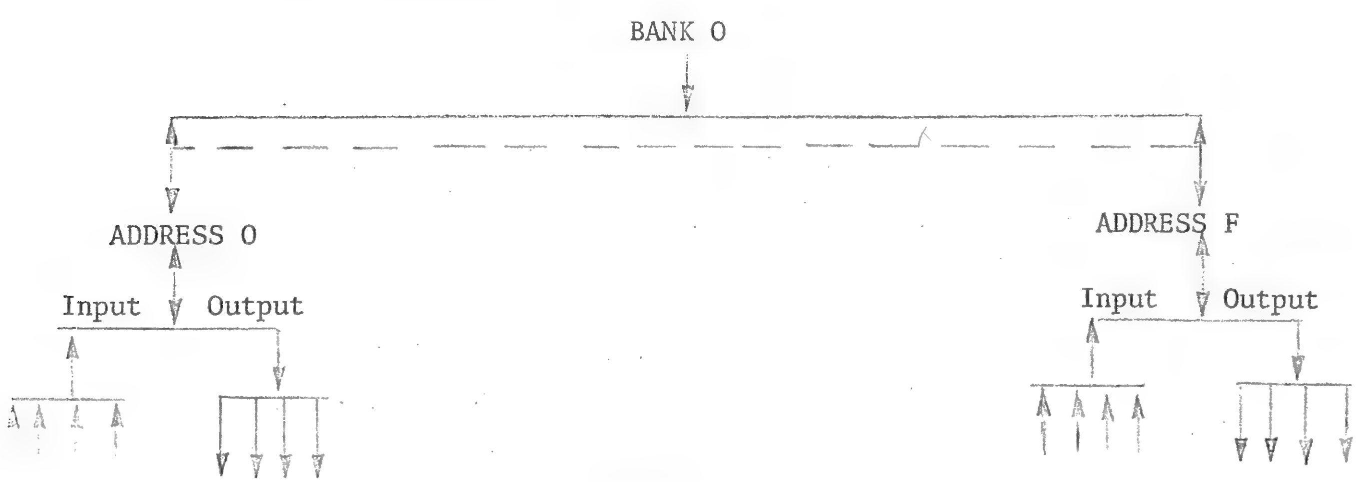


Figure 2.6.1

There are 0 to F banks and 0 to F addresses in each bank and four lines of input and output for each address.

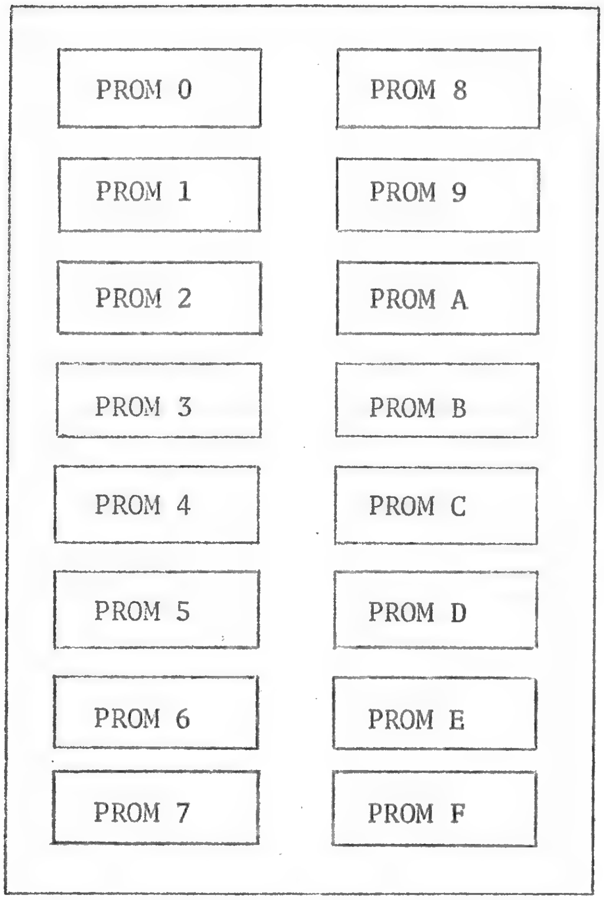
Figure 2.6.2

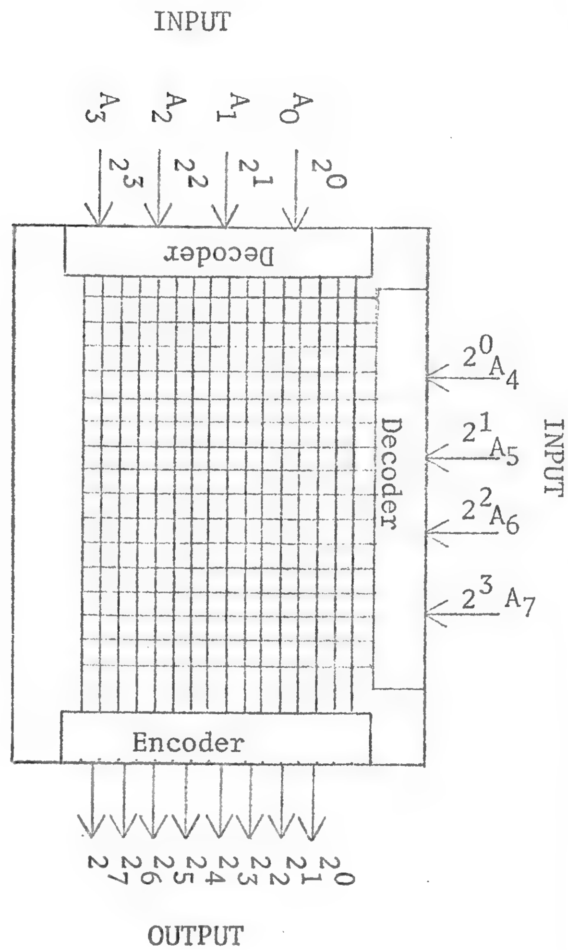
Absolute Address range (HEX)	MEMORY BANK (Hexadecimal)	Relative Address (HEX)
FF FO	F	F 0
EF EO	E	F 0
DF DO	D	F 0
CF CO	C	F 0
BF BO	B	F 0
AF AO	A	F 0
9F 90	9	F 0
8F 80	8	F 0
7F 70	7	F 0
6F 60	6	F 0
5F 50	5	F 0
4F 40	4	F 0
3F 30	3	F 0
2F 20	2	F 0
1F 10	1	F 0
OF 00	0	F 0

Ascending Addressing from Least Significant to Most Significant Address

I/O Addressing Scheme

PROM Module Addressing:



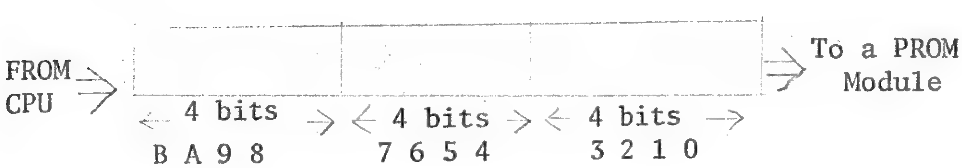


As shown above:

There are maximum of 16 PROM Chips

Each PROM chip has 256 x 8 bit locations

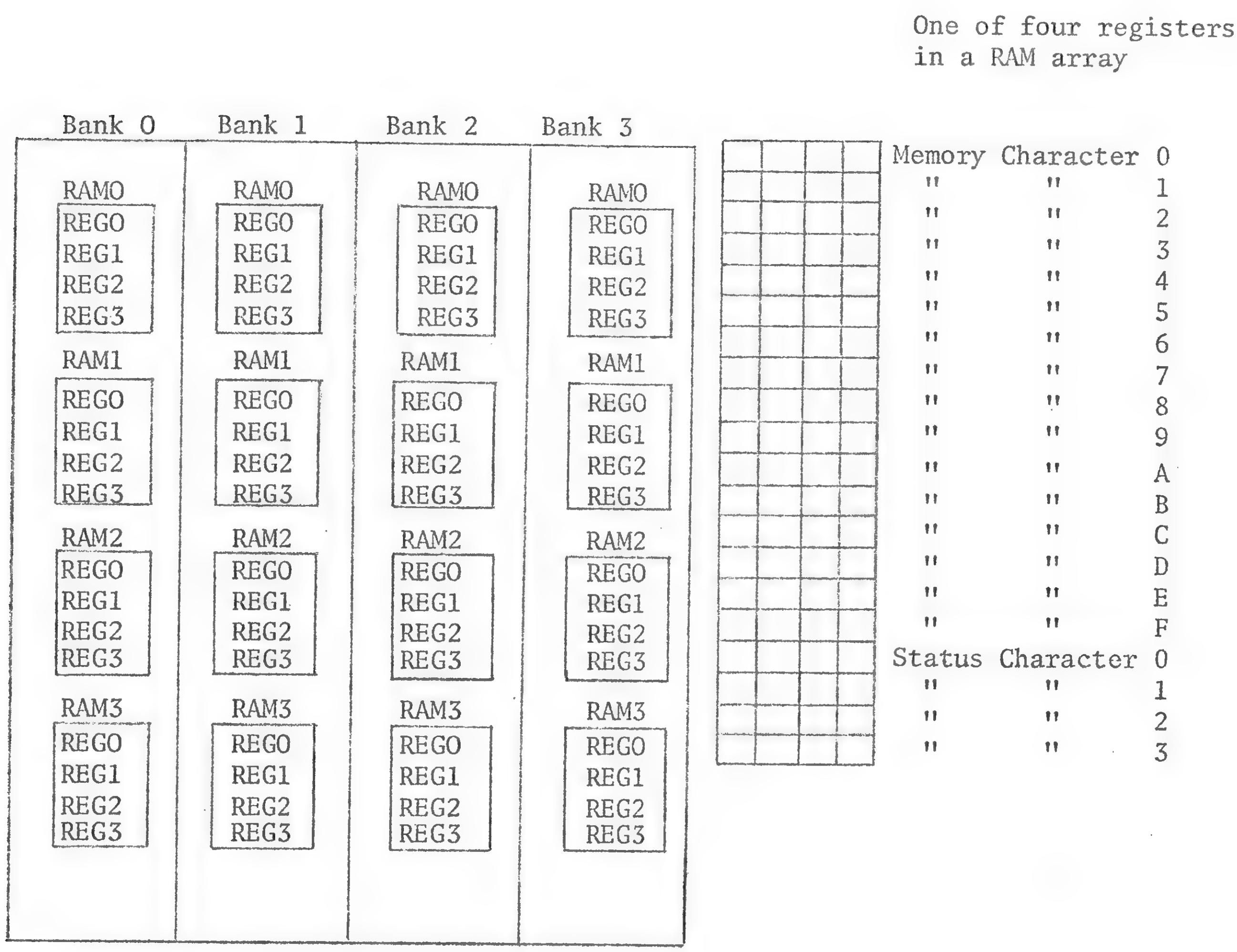
CPU sends 12 (or 8) bits of address — 4 bits at a time to a PROM Module.



The last four bits of address (8 thru B_{16}) selects one of 16 chips and the first eight bits (0 thru 7 bits) select one of 256 locations within the PROM chip.

RAM MODULE ADDRESSING

Version A



As shown above:

There are maximum of 4 banks
Each Bank has 4 RAM chips
Each RAM has 4 registers

To address a register:

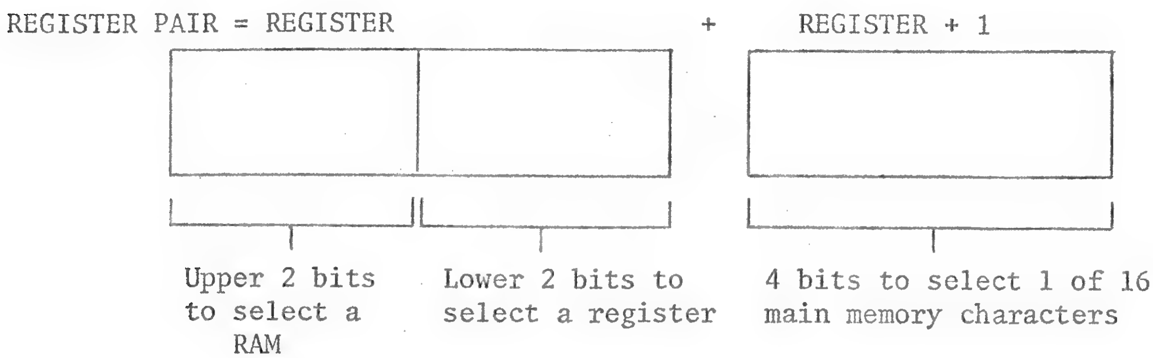
1st Select RAM bank code (0, 1, 2, 4) utilizing DCL instruction (Load Bank, number in the accumulator before using DCL instruction)

2nd Select RAM within the Bank

3rd Select a register within the RAM

4th Select character of Main Memory if applicable

With the help of software, load register pair with appropriate values.



'Register' Contents Upper 2 bits	Lower 2 bits	DESCRIPTION
00	00	RAM0, Register 0
00	01	RAM0, Register 1
00	10	RAM0, Register 2
00	11	RAM0, Register 3
01	00	RAM1, Register 0
01	01	RAM1, Register 1
01	10	RAM1, Register 2
01	11	RAM1, Register 3
10	00	RAM2, Register 0
10	01	RAM2, Register 1
10	10	RAM2, Register 2
10	11	RAM2, Register 3
11	00	RAM3, Register 0
11	01	RAM3, Register 1
11	10	RAM3, Register 2
11	11	RAM3, Reigster 3

Each RAM chip has four output ports. They can be accessed by selecting a Bank and selecting a RAM chip within the Bank.

Version B

This module is twice as big as Version A. Addressing of this module is basically the same. However, unlike Version A, the selection of a bank is made according to the following truth table.

Accumulator	Bank No.
000	0
001	1
010	2
100	3
011	4
101	5
110	6
111	7

Truth Table

NOTE: The first four codings of Version B is the same as the Version A's coding.

2.7 Module Mounting

Depending upon number of modules, cabinet selection can be made and necessary information is provided to make the selection.

2.7.1 Open Chassis Mounting

2.7.1.1 Single Open Chassis Mounting (9007-0300)
The single open chassis rack furnishes space for a maximum of one small Power Supply and twelve data control bus modules or without the Power Supply there can be up to sixteen modules. This standard rack (size 18" x 7½" x 6¼") is provided with two terminal strips allowing for 42 terminations but can optionally be modified to include four terminal strips with 84 terminations. This rack provides all necessary hardware to insert modules per correct spacing.

2.7.1.2 Double Open Chassis Mounting (9007-0400)
The double open chassis rack furnishes space for a maximum of one small Power Supply and 28 data control bus modules or without the Power Supply there can be up to thiry-two modules. This rack (size 18" x 12.8" x 6¼") is provided with four terminal strips allowing for 84 terminations. This rack provides all necessary hardware to insert modules per correct spacing.

OPEN CHASSIS MOUNTING IS IDEAL FOR USES WHERE THE MICROCOMPUTER IS USED AS A MACHINE CONTROLLER AND IS MOUNTED IN A NEMA TYPE BOX.

2.7.1.3 Cable Rack(9007-0410) This is designed for mounting either up to eleven (11) 50 conductor telephone type connectors or up to twenty-two (22) 14 pin terminal strips. The cable rack is particularly useful in applications where input and output lines exceed open chassis (single or double) capacity.

2.7.2 Rack Mounting

2.7.2.1 Single Rack Mounting(9007-P300) The Comstar 4 can be built in an enclosure to mount in the standard 19" rack configuration. The dimension of this enclosure is 19" x 8 3/4" x 9",or 9007-P350 is 24" wide.

2.7.2.2 Double Rack Mounting(9007-P400) This enclosure encompasses the double panel mount for installation in the standard 19" x 14" x 9",or 9007-P450 is 24" wide.

3.2.3 Cable Rack - This can be mounted if more connections are needed. Refer to 3.1.3

2.7.3 Table Top Mounting

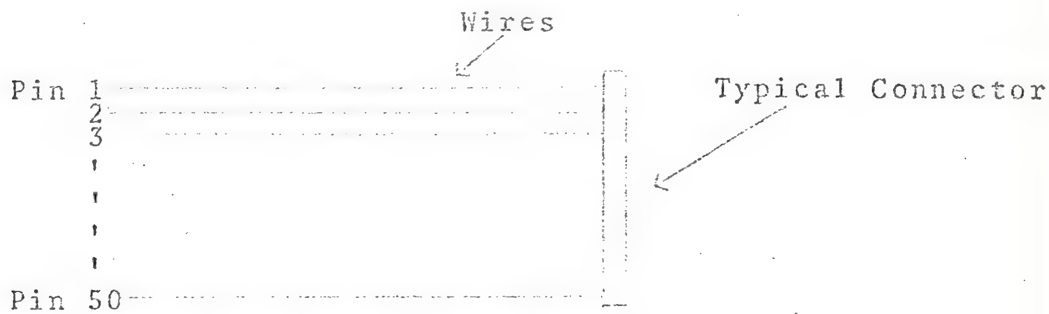
Comstar 4 microcomputer can be mounted in a table top cabinet for convenience. There are two types of table top mountings - single row for single rack mounting and double for double rack mounting. Rear cable panel is available for internal/external connections.

2.7.4 Card Guides and Edge Connectors

Steel cadmium plated card guides with A.B.S. plastic insulation are properly sized for CS4 modules and are provided with standard slot spacing of one inch (25.40mm). Optional spacing of ½ inch (12.70mm) and 3/4 inch (19.05mm) are available per special order. A 22/44 (pair/contacts) edge type connector with gold contacts and necessary polarizing keys are provided with mounting brackets and screws.

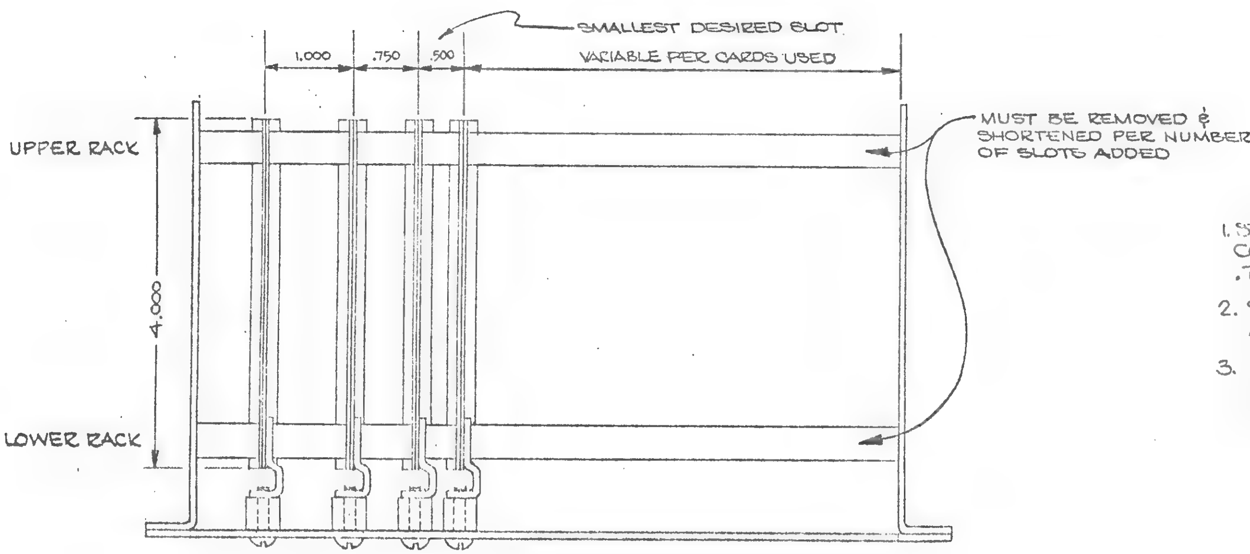
2.7.5 50 Conductor Flat Cable (D/C Bus)

The cable consists of 50 wires and desired number of connectors. This flat cable eliminates open wire problems, cross-connector shorts and loose contacts. Each connector in the flat cable is constructed with alignment grooves to insure proper positioning of the cable. Comstar suggests a minimum space of 1 inch between the connectors.



SPACER SIZE		
SPACE	UPPER RACK	LOWER RACK
1.000	.950	.888
.750	.700	.638
.500	.450	.388
Optional	.050 Less than space req'd.	.112 Less than space req'd.

REVISIONS					
REV	ECO	DESCRIPTION	DRN	DATE	APP



NOTES

1. STANDARD SPACING FOR CARD SLOTS IN THE COMSTAR CS4 COMPUTER RACK ARE 1.00, .750, & .500
2. STANDARD RACKS PER CUSTOMER REQUEST ARE ASSEMBLED BY COMSTAR
3. FIELD CHANGES BY THE CUSTOMER CAN BE MADE BY ORDERING GUIDE PACKAGES FROM COMSTAR & INSTALLING THEM PER THIS PRINT.

COMSTAR
7413 WASHINGTON AVENUE SOUTH, EDINA, MINNESOTA

TITLE MECHANICAL ASSEMBLY FOR COMSTAR CS4 RACK

DRAWN H. JOHNSON 10-2-73	SCALE	SH OF	
CHECKED	SIZE C	DRAWING NO.	REV.
APPROVED			

CHAPTER 3 Machine Language Programming

3.1 General Indtroduction

3.2 Electrically Programmable ROMS

3.3 Programming the SPROM/FPROM

3.4 Programmers

3.5 Erasing the SPROM/FPROM

3.6 Machine Language Programming

- 3.6.1 Comstar 4 System Organization
- 3.6.2 Instruction Table
- 3.6.3 Detailed Instruction Set

3.7 Flowchart Symbols and Guidelines

3.8 Programming Examples

- 3.8.1 Addition
- 3.8.2 Paper Tape Punch
- 3.8.3 Output of Data to Typesetting Punch
- 3.8.4 Pulse Generator
- 3.8.5 Multiplication
- 3.8.6 Division

CHAPTER 3

3.1 General Introduction:

To store microprograms, the PROM has to be programmed and the machine language programming is the basic method of programming. The machine language user is expected to have a sound understanding of the Comstar 4 microcomputer organization and the significance of the instruction set. The machine language is the most efficient method to use the PROM chip (opposed to Comstar's Process Control Language). It is also the easiest method to write programs with stringent timing requirements.

3.2 ELECTRICALLY PROGRAMMABLE ROMS

Comstar customers use two types of Programmable Read Only Memory (PROM) Chips, Slow Programmable Read Only Memory (SPROM) and Fast Programmable Read Only Memory (FPROM).

The SPROM and FPROM are 256 by 8 bit word electrically Programmable Read Only Memories (PROMs) ideally suited for uses where fast turn-around and pattern experimentation are important. The SPROM and FPROM are packaged in a 24 pin dual in-line package with a transparent quartz lid. The transparent quarts lid allows the user to expose the chip to ultraviolet light to erase the bit pattern. Therefore, unlike a metal mask Read-Only-Memory where a pattern cannot be changed, a new pattern can be written in to the PROM devices.

There are no electrical differences between the FPROM and the SPROM in the read mode. However, the FPROM offers a faster programming time than the SPROM. The FPROM chip's waiting time is 2 minutes versus 20 minutes for the SPROM.

3.3 PROGRAMMING THE SPROM-FPROM

As supplied or erased, all data bits in SPROM are initially interpreted as zeroes (output high), with the programming operation forcing zeroes to ones or leaving zeroes unchanged. Ultraviolet erasure of the SPROM restores the data to all zeroes.

3.4 PROM PROGRAMMING EQUIPMENT

All of the available equipment is discussed in Section 2.2.3.

3.5 ERASING THE SPROM/FPROM

The SPROM and FPROM Chips may be erased by exposure to high intensity short-wave ultraviolet light at a wavelength of 2537 Å. The recommended integrated dose (i.e., UV intensity x intensity time) is 6W-sec/cm^2. The devices are made with a transparent quartz lid covering the silicon die. Conventional room light, fluorescent light, or sunlight has no measurable effect on stored data, even after years of exposure. However, after 10 to 20 minutes under a suitable Ultraviolet concentrated light source, the device is erased to a state of all zeros (SPROM) or all ones (FPROM). It is recommended that no more ultraviolet light exposure than that necessary to erase the SPROM/FPROM should be used to prevent damage to the device.

CAUTION: WHEN USING AN ULTRAVIOLET SOURCE OF THIS TYPE, ONE SHOULD BE CAREFUL NOT TO EXPOSE ONE'S EYES OR SKIN TO THE ULTRAVIOLET RAYS BECAUSE OF THE DAMAGE TO VISION, OR BURNS WHICH MIGHT OCCUR. IN ADDITION, THESE SHORTWAVE RAYS MAY GENERATE CONSIDERABLE AMOUNTS OF OZONE WHICH IS ALSO POTENTIALLY HAZARDOUS.

3.6 MACHINE LANGUAGE PROGRAMMING

3.6.1 COMSTAR 4 System Organization

CENTRAL PROCESSING UNIT

Central processing unit consist of a central processing unit (CPU) and a memory that has a stored sequence of instructions for the CPU. The CPU is operated by a clock circuit to alternately fetch and execute the memory instructions. The CPU fetches an instruction by sending an address from a program address counting register to the program memory. The program memory decodes the address and sends the selected instruction to the CPU. The CPU stores the instruction in an instruction register where it is decoded and executed.

COMSTAR 4 Systems

The COMSTAR 4 Systems are controlled by the Intel 4004 CPU Chip. The CPU performs control and data transfer functions with the logic elements shown in the system data flow diagram Figure 3.6.1.1. The CPU communicates with program memory, RAM registers and I/O ports by connecting appropriate elements of the system to the 4 bit CPU BUS. Conceptually the information paths exist as shown in Figure 3.6.1.1.

In addition to an instruction register and program address counter, the CPU contains a program address counter stack, an arithmetic logic unit (ALU) with a four bit accumulator register, and 16 four bit registers for intermediate data storage.

INSTRUCTION REGISTER

The instruction register shown in Figure 3.6.1.2 consists of eight bits of storage and decoding for single word and the first word of double word instructions as they are received from the program memory. The 4 bits associated with M1 time are always instruction information. The 4 bits at M2 time can be additional instruction information, data constants, or page address information.

The second word of a double word instruction does not go to the instruction register but goes as either data to the index registers or as a word address to A1 and A2 of the program address counter.

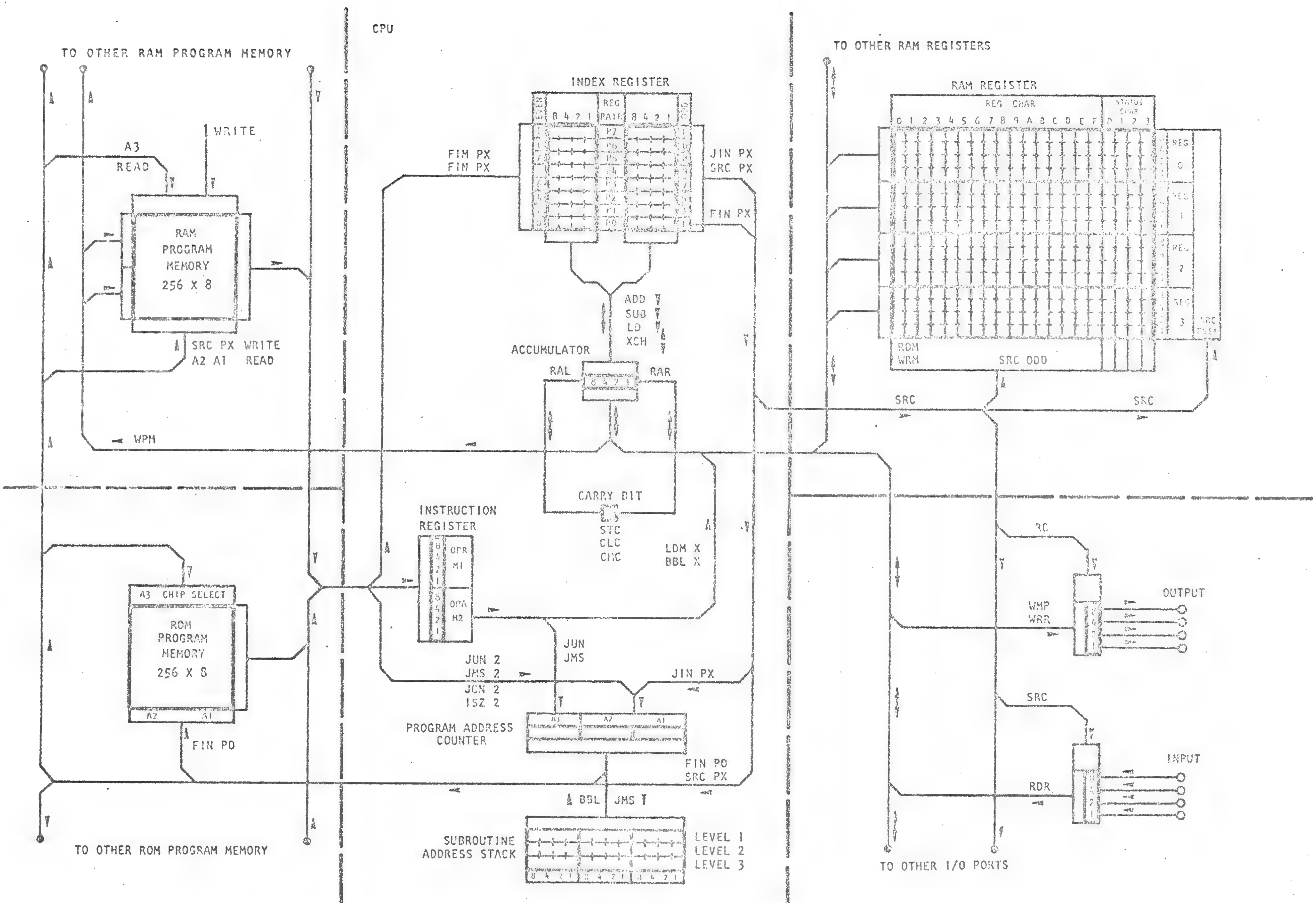


FIGURE 3.6.1.1

COMSTAR 4 System Data Flow

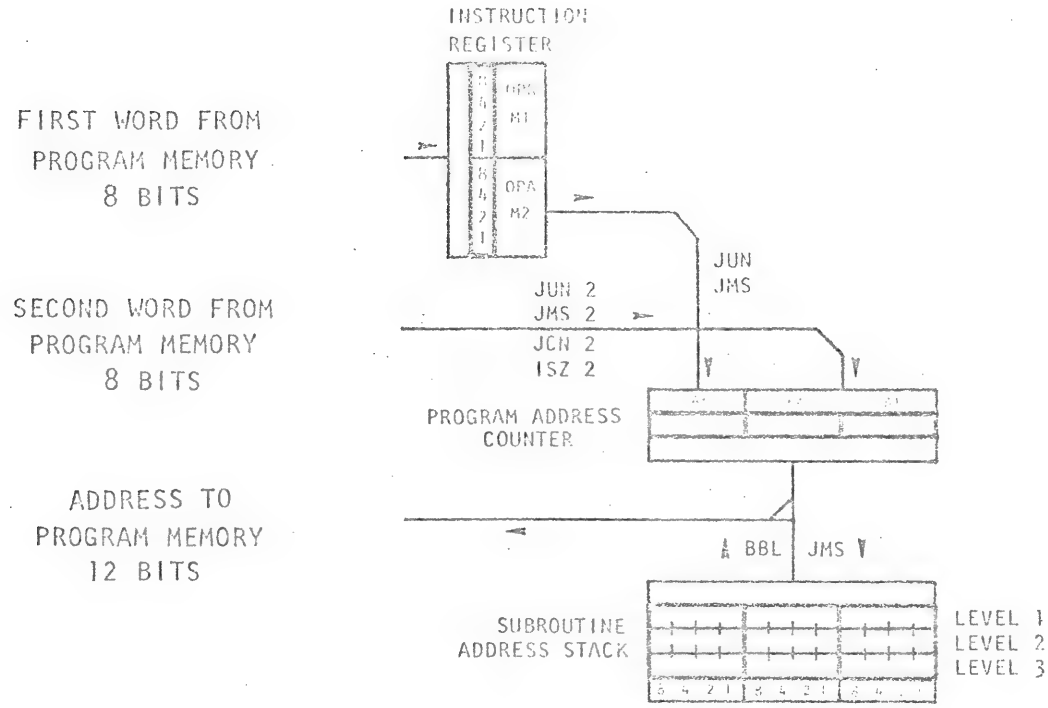


FIGURE 3.6.1.2

Instruction Register, Program Address Counter, and Subroutine Address Stack

PROGRAM ADDRESS COUNTER

The program address counter shown in Figure 3.6.1.2 is a 12 bit sequential counter which keeps track of the location of the next instruction to be executed from program memory. The four most significant bits (A3) are called the page address and the eight least significant bits (A2 and A1) are the word address of the instruction on a page. The program address counter is normally incremented by 1 for each instruction word unless the instruction is the type which modifies the count by loading a new address.

SUBROUTINE ADDRESS STACK

The subroutine address stack shown in Figure 3.6.1.2 consists of three 12 bit registers used to save the program return address for each of three allowable subroutine levels. The subroutine address stack is controlled by two CPU instructions, an entry instruction JMS and a return instruction BBL. Each entry to a subroutine causes the program address counter to be transferred to the top most level of the subroutine address stack. The three levels in turn are pushed-down to accommodate the new entry. The lowest level is lost off the bottom of the stack. Each return from a subroutine causes the stack to be pulled-up one level with the top most address going to the program address counter.

INDEX REGISTERS

The index registers consist of sixteen 4 bit registers which can be directly operated on by various instructions, either individually or in pairs. Figure 3.6.1.3 shows the registers organized as the even numbered and the odd numbered registers, or as seven pairs, each pair consisting of one even and one odd numbered register.

When the registers are being used with the 4 bit accumulator by various instructions they are used individually. When data is loaded direct from program memory or the registers are used for address control they are used in pairs because of the 8 bit requirement for these functions.

ARITHMETIC LOGIC UNIT

The arithmetic logic unit consists of a 4 bit accumulator register and a carry flip-flop as shown in Figure 3.6.1.3. In addition to providing the arithmetic functions of ADD and SUBtract the accumulator is the central control and distribution point for data flow in the system. All data transfers to and from I/O. RAM registers and the index register occurs with the accumulator register.

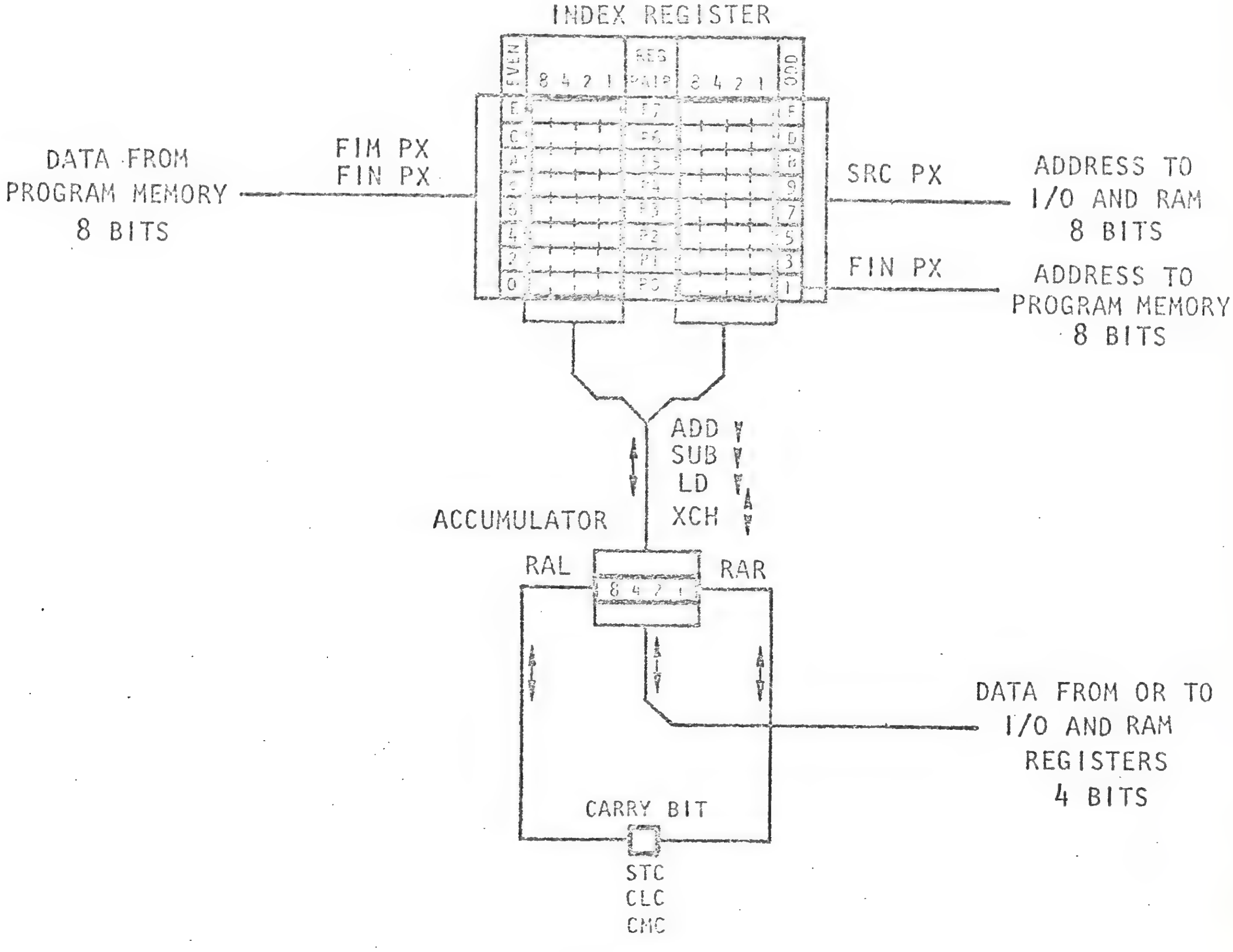


FIGURE 3.6.1.3

Index Registers

In addition to the instructions which control data transfer to or from the accumulator there are instructions which directly control the accumulator or its associated carry bit. The accumulator can be tested, incremented, decremented, set to any value, cleared, complemented, rotated right or left through the carry besides being manipulated for decimal arithmetic. The carry bit can be set, cleared, complemented, or tested.

PROGRAM MEMORY

Program memory stores the instruction to be executed by the CPU and is defined by the CPU instruction set as a page oriented memory of 256 words per page as shown in Figure 3.6.1.4. The CPU addresses the page and word and the program memory sends the 8 bit word at that address to the CPU.

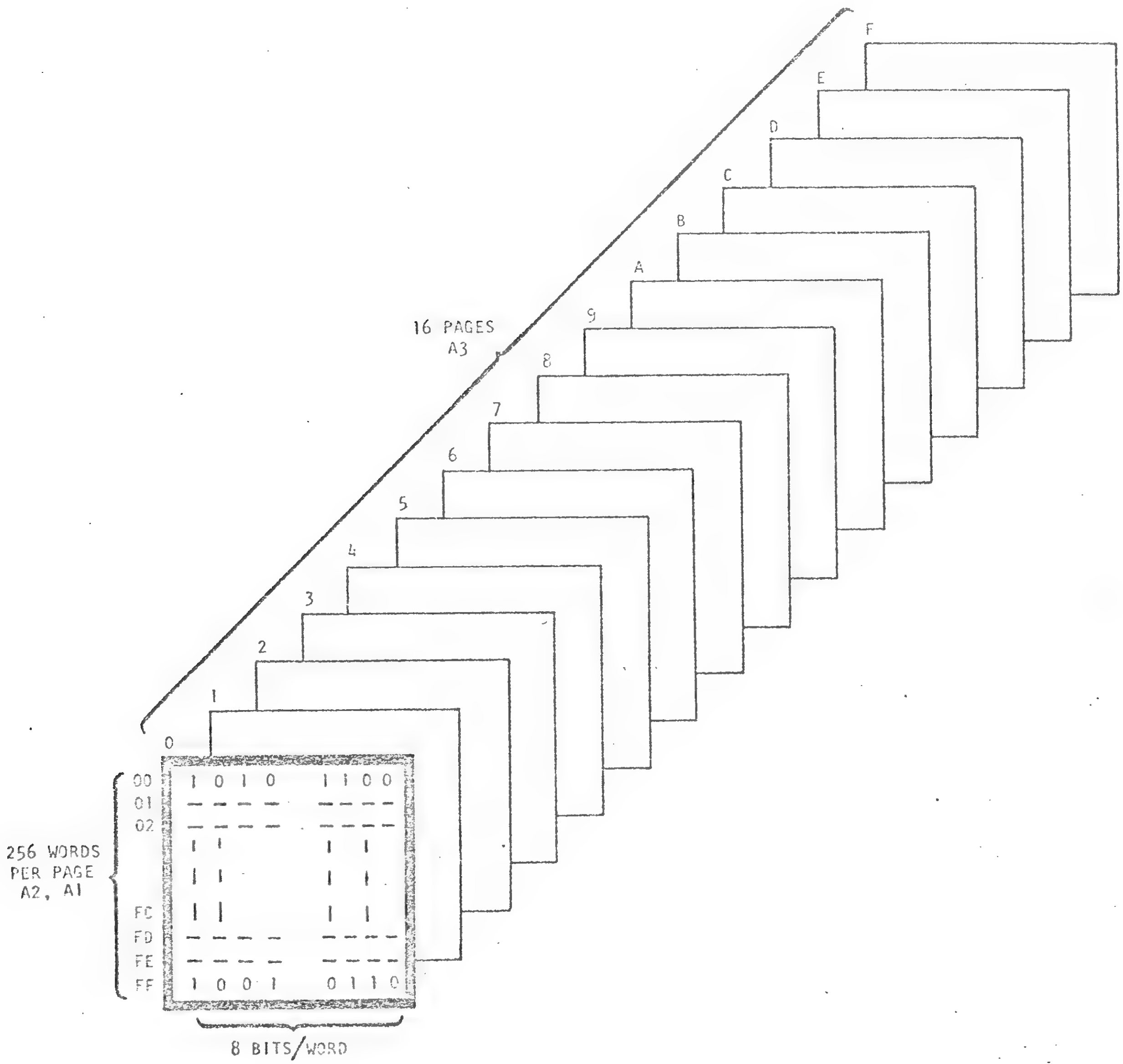


FIGURE 3.6.1.4

Organization of Program Memory as Defined by the CPU Instructions

The 12 bit addressing capability of the CPU allows direct access to 16 pages with the four A3 bits used as page address. The eight bits at A1 and A2 are used for the word address within a page. It is important to understand the page organization in terms of the address control instructions (jumping and branching). Certain address instructions use the full twelve bits of address and may be used to change control within a page or from page to page. Other address control instructions use only eight bits of address and are limited to changing control only within a page.

The COMSTAR 4 systems are implemented with ROM (read only memory) program memory only. In addition the CS-PROM has all the control lines available for implementing RWM (read write memory) program memory. ROM program memory is used for systems in fixed applications. RWM memory is used where it is desired to change the system application by the operator. RWM is a considerable step in system complexity in hardware and programs, and is therefore not recommended unless absolutely necessary.

ROM PROGRAM MEMORY

ROM program memory on the COMSTAR 4 system is accomplished as shown in Figure 3.6.1.4 using programmable erasable ROMs organized as 256 location of 8 bits. This organization is equivalent to the page size of the CPU therefore each ROM chip equals one page. Other ROM sizes and organizations can be used if the appropriate hardware addressing is provided.

ROM program memory addressing is an automatic function of the C4 hardware. The only control the program designer has over ROM memory is use of the program control instructions to change the instruction sequence.

RAM REGISTER STORAGE

The COMSTAR 4 systems use the Intel 4002 RAM register devices for program controlled data storage. Each 4002 is organized as four registers of 20 characters as shown in Figure 3.6.1.5. Each 20 character register consists of 16 individually addressable characters of main storage plus 4 instruction selectable status characters.

The instruction capability of the CPU allows addressing of up to 32 of the 4002 RAM devices. This is accomplished through an organization of 8 banks of 4 RAM chips per bank. RAM banks are selected by the DCL instruction that specifies which of the four CM-RAM lines out of the CPU will be active. The active CM-RAM line designates which RAM bank will respond to the SRC instruction. The SRC instruction selects the RAM chip, register and character. A summary of RAM addressing is given in Table 3.6.1.1 and further definition of RAM addressing is given in Section 6 under the SRC and DCL instructions.

TABLE 3.6.1.1

RAM Addressing

Level	Instruction
RAM Bank	DCL
RAM Chip	SRC, Even Register high order bits
RAM Register	SRC, Even Register low order bits
RAM Character	SRC, Odd Register 4 bits

INPUTS AND OUTPUTS

The flow of data into and out of the C4 systems is accomplished through the I/O ports of four lines each. To accomplish an input or output function a port must first be addressed by the CPU instruction SRC. The even register of the SRC pair contains the address of the port to be selected. Once a port has been addressed it remains selected for as many input or output operations as desired until another port is addressed.

There are two types of output ports and one type of input port. Each RAM register device has an output port packaged physically within the device. This port shares chip select addressing with the RAM but has its own instruction WMP for the transfer of data from the accumulator to the port. The port latches any data sent to it and retains it as a stable output until a subsequent WMP instruction changes the data. The RAM port lines are MOS low level active outputs capable of driving one low power TTL load.

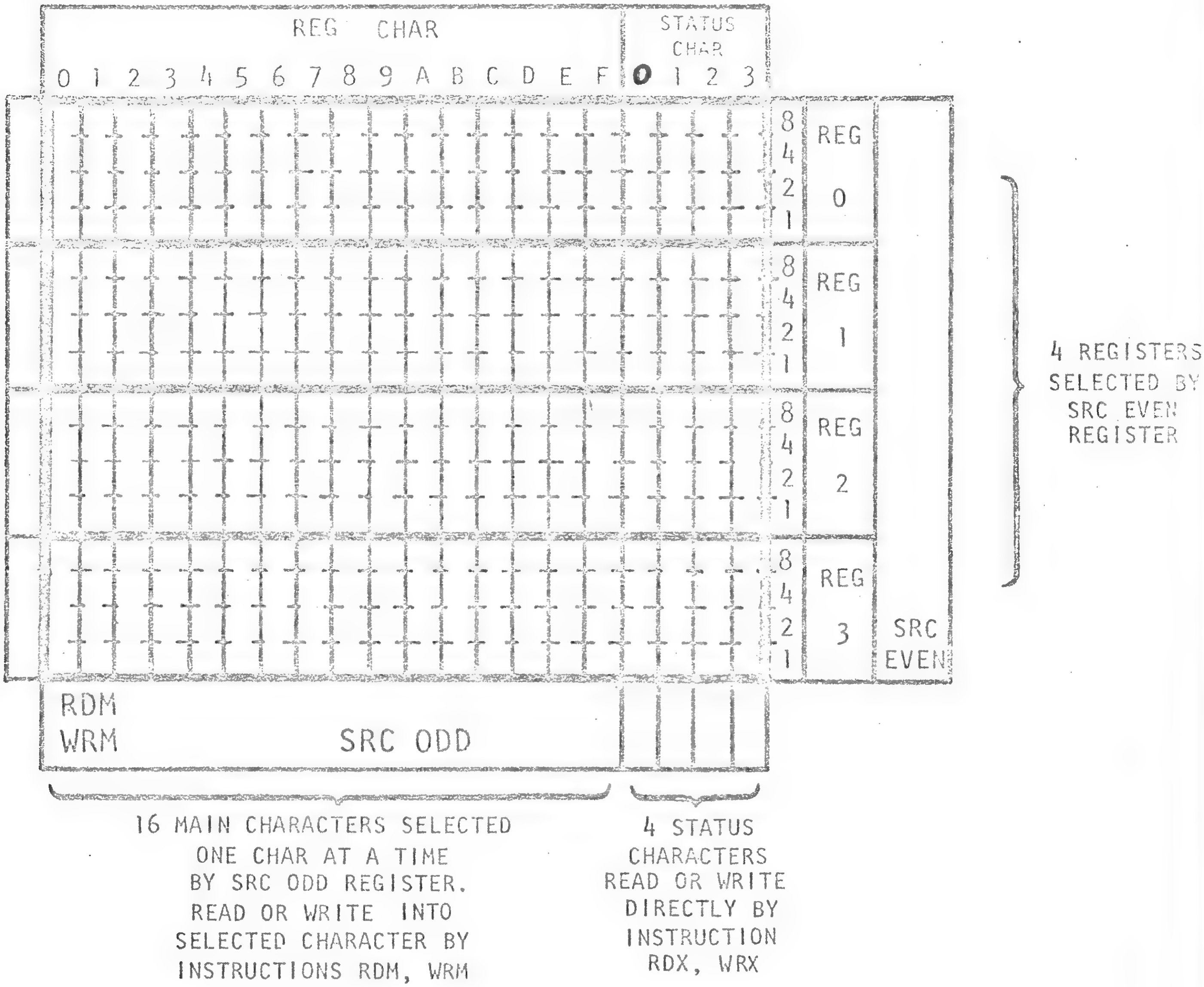


FIGURE 3.6.1.5

RAM Index Register

The other type of output port is implemented in the COMSTAR 4 systems using TTL logic. The CPU instruction WRR is used to send data to a TTL quad D type flip-flop from the accumulator. The TTL flip-flops latch the data as a stable output until a subsequent WRR instruction changes the data.

The COMSTAR 4 input ports are also implemented with TTL logic. The CPU instruction RDR reads data from the selected input port into the accumulator.

HEXADECIMAL NOTATION

The basic 4 bit structure of the CPU makes it convenient to use hexadecimal notation to express with a single character, one-of-sixteen possible combinations.

The single hexadecimal character notation 0 ⟶ 9, A ⟶ F is used to refer to the:

- 16 Basic Instructions
- 16 I/O and RAM Instructions
- 16 Accumulator Instructions
- 16 Index Registers
- 16 Pages of Program Memory Capacity
- 16 RAM Register Chip Capacity
- 16 Characters in a RAM Register
- 16 Output Ports
- 16 Input Ports

A double hexadecimal character notation is applied to the 8 bit instruction word address for program memory, where the decimal addresses 000 through 255 are given as 00 through FF in hexadecimal.

Table 3.6.1.2 shows the hexadecimal notation for sixteen combinations. Additional hexadecimal tables are given in the appendix.

TABLE 3.6.1.2

Hexadecimal Notation for Sixteen Combinations

Hexadecimal	Binary	Decimal
0	0000	0
1	0001	1
2	0010	2
3	0011	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11
C	1100	12
D	1101	13
E	1110	14
F	1111	15

3.6.2 INSTRUCTION TABLE

This section presents the 4004 CPU instructions in a short table form. Section 6 contains detailed descriptions of the instructions.

HEX CODING		MNEMONIC OPR	MNEMONIC OPA	DESCRIPTION OF OPERATION
0	0	NOP		No operation.
1 A_2	C_x A_1	JCN	C_x LABEL	Jump on condition C_x to the program memory address A_1, A_2, otherwise continue in sequence. (see back cover).
2 D_2	P_x0 D_1	FIM D_2	P_x D_1	Fetch immediate from program memory data D_1, D_2 to index register pair P_x
2	P_x1	SRC	P_x	Send register control. Send the contents of index register pair P_x to I/O ports and RAM register as chip select and RAM character address.
3	P_x0	FIN	P_x	Fetch indirect. Send contents of register pair 0 out as a program memory address. Data fetched is placed into register pair P_x
3	P_x1	JIN	P_x	Jump indirect. Jump to the program memory address designated by contents of register pair P_x
4 A_2	A_3 A_1	JUN	LABEL	Jump unconditional to program memory address A_1, A_2, A_3.
5 A_2	A_3 A_1	JMS	LABEL	Jump to subroutine located at program memory address A_1, A_2, A_3. Save previous address (push down in stack).
6	R_x	INC	R_x	Increment contents of register R_x.
7 A_2	R_x A_1	ISZ	R_x LABEL	Increment and step on zero. Increment contents of register R_x, if result is not 0 go to program memory address A_1, A_2, otherwise step to the next instruction in sequence.
8	R_x	ADD	R_x	Add contents of register R_x to accumulator.
9	R_x	SUB	R_x	Subtract contents of register R_x to accumulator with borrow.
A	R_x	LD	R_x	Load contents of register R_x to accumulator.
B	R_x	XCH	R_x	Exchange contents of index register R_x and accumulator.
C	D_x	BBL	D_x	Branch back one level in stack to the program memory address stored by a prior JMS instruction. Load data D_x to accumulator.
D	D_x	LDM	D_x	Load data D_x to accumulator.
E	X	I/O and RAM register instructions		
F	X	Accumulator instructions		

A_1 Low order address bits
A_2 High order address bits
A_3 Chip select

P_x1 Register pairs P_0 through P_7 designated by odd characters 1, 3, 5, 7, 9, B, D, F
P_x0 Register pairs P_0 through P_7 designated by even characters 0, 2, 4, 6, 8, A, C, E
R_x Register 0 ⟶ F

D_x Data
D_1 Data for odd register
D_2 Data for even register
C_x Jump conditions

I/O AND RAM REGISTER INSTRUCTIONS

HEX CODING	MNEMONIC OPR	OPA	DESCRIPTION OF OPERATION
E 0	WRM		Write the contents of the accumulator into the previously selected RAM register character.
E 1	WMP		Write the contents of the accumulator into the previously selected RAM output port. (Output lines.)
E 2	WRR		Write the contents of the accumulator into the previously selected output port. (I/O lines.)
E 3	WPM		Write the contents of the accumulator into the previously selected RAM program memory.
E 4	WR0		Write the contents of the accumulator into the previously selected RAM status character 0.
E 5	WR1		Write the contents of the accumulator into the previously selected RAM status character 1.
E 6	WR2		Write the contents of the accumulator into the previously selected RAM status character 2.
E 7	WR3		Write the contents of the accumulator into the previously selected RAM status character 3.
E 8	SBM		Subtract the previously selected RAM register character from accumulator with borrow.
E 9	RDM		Read the previously selected RAM register character into the accumulator.
E A	RDR		Read the contents of the previously selected input port into the accumulator. (I/O lines.)
E B	ADM		Add the previously selected RAM register character to accumulator with carry.
E C	RD0		Read the previously selected RAM status character 0 into accumulator
E D	RD1		Read the previously selected RAM status character 1 into accumulator.
E E	RD2		Read the previously selected RAM status character 2 into accumulator.
E E	RD3		Read the previously selected RAM status character 3 into accumulator.

ACCUMULATOR INSTRUCTIONS

HEX CODING	MNEMONIC OPR	OPA	DESCRIPTION OF OPERATION
F 0	CLB		Clear both. (Accumulator and carry.)
F 1	CLC		Clear carry.
F 2	IAC		Increment accumulator.
F 3	CMC		Complement carry.
F 4	CMA		Complement accumulator.
F 5	RAL		Rotate left. (Accumulator and carry.)
F 6	RAR		Rotate right. (Accumulator and carry.)
F 7	TCC		Transmit carry to accumulator and clear carry.
F 8	DAC		Decrement accumulator.
F 9	TCS		Transfer carry subtract and clear carry.
F A	STC		Set carry.
F B	DAA		Decimal adjust accumulator.
F C	KBP		Keyboard process. Converts the contents of the accumulator from a one out of four code to a binary code.
F D	DCL		Designate command line.
F E			
F F			

C_X CONDITION TABLE FOR JCN INSTRUCTION

JCN HEX	C_X MNEMONIC	C_8	C_4	C_2	C_1	C_8: Invert Jump Condition C_4: Jump if Accumulator = 0 C_2: Jump if Carry Bit = 1 C_1: Jump if Test Input = 0 (High)
10		0	0	0	0	NO OPERATION
11	TO	0	0	0	1	Jump if test = 0 (High)
12	C1	0	0	1	0	Jump if CY = 1
13	TO+C1	0	0	1	1	Jump if test = 0 or CY = 1
14	AO	0	1	0	0	Jump if AC = 0
15	TO+AO	0	1	0	1	Jump if test = 0 or AC = 0
16	C1+AO	0	1	1	0	Jump if CY = 1 or AC = 0
17	TO+C1+AO	0	1	1	1	Jump if test = 0 or CY = 1 or AC = 0
18		1	0	0	0	Jump Unconditionally
19	T1	1	0	0	1	Jump if test = 1 (Low)
1A	CO	1	0	1	0	Jump if CY = 0
1B	T1CO	1	0	1	1	Jump if test = 1 and CY = 0
1C	A1	1	1	0	0	Jump if AC ≠ 0
1D	T1A1	1	1	0	1	Jump if test = 1 and AC ≠ 0
1E	COA1	1	1	1	0	Jump if CY = 0 and AC ≠ 0
1F	T1COA1	1	1	1	1	Jump if test = 1 and CY = 0 and AC ≠ 0

REGISTER PAIR PX LOOKUP TABLE

	PX 0 RRR 0		PX 1 RRR 1	
PX	FIM	FIN	SRC	JIN
P0	20	30	21	31
P1	22	32	23	33
P2	24	34	25	35
P3	26	36	27	37
P4	28	38	29	39
P5	2A	3A	2B	3B
P6	2C	3C	2D	3D
P7	2E	3E	2F	3F

3.6.3 INSTRUCTION DESCRIPTIONS

NO OPERATION NOP

M1				M2			
0	0	0	0	0	0	0	0
8	4	2	1	8	4	2	1

0_0

No operation is performed by this instruction except that the program address counter counts to the next instruction address in sequence. This instruction can be used as a one cycle time delay. To avoid problems with power-on reset, the first instruction at program address 000 should always be an NOP.

JUMP ON CONDITION JCN

M1				M2			
0	0	0	1	C_x			
8	4	2	1	8	4	2	1

First Word $1\ C_x$

A_2				A_1			
8	4	2	1	8	4	2	1

Second Word $A_2\ A_1$

If the designated condition (C_x) is true, program control is transferred to the instruction located at the 8 bit address A_2, A_1 of the current page, otherwise program control continues in sequence. If the JCN occupies the last two positions of a page or overlaps the page boundary, program control is transferred to the 8 bit address on the next page in sequence.

JCN is one of the two decision making instructions of the CPU, the other being ISZ. JCN allows a decision on the following tests:

Test accumulator for zero or nonzero

Test carry bit for logic one or zero

Test external input lead for high or low

Table 3.6.2 provides detailed definitions of conditions C_x.

FETCH IMMEDIATE

FIM

M1				M2			
0	0	1	0	R	R	R	0
8	4	2	1	8	4	2	1

First Word — $2\ P_x0$

D_2				D_1			
8	4	2	1	8	4	2	1

Second Word — $D_2\ D_1$

Load the 8 bits of data from the second word D_2, D_1, to the designated pair of index registers P_x0 where D_2 is data for the even register and D_1 is data for the odd register.

FIM uses the even register numbers to designate a pair. The only valid operand codes for P_x0 are 0, 2, 4, 6, 8, A, C, and E. FIM provides the most efficient way to initialize a pair of index registers.

RRR defines one of the eight register pairs P0 through P7. The 0 following RRR is part of the command decoding and distinguishes the FIM from the SRC.

SEND REGISTER CONTROL

SRC

M1				M2			
0	0	1	0	R	R	R	1
8	4	2	1	8	4	2	1

$2\ P_x1$

Send the contents of index register pair P_x1 to the I/O ports and RAM registers as chip select and/or RAM character select. SRC uses the odd register numbers to designate a pair. The only valid operand codes for P_x1 are 1, 3, 5, 7, 9, B, D, and F.

RRR defines one of the eight register pairs P0 through P7. The 1 following RRR is part of the command decoding and distinguishes the SRC from the FIM.

It is necessary to address the I/O port or RAM register character using an SRC instruction before an I/O operation or a RAM register operation can be performed. The same SRC instruction can be used to address both I/O ports and RAM registers, however, the meaning of the address in the designated pair P_x1 is different for each as shown below.

The I/O port is addressed by the contents of the even register designated by P_x. The odd register does not serve any purpose in selecting I/O ports.

ELEMENT ADDRESSED	REGISTER PAIR P_X	
	EVEN REGISTER	ODD REGISTER
I/O PORT	PORT SELECT 8 4 2 1	~~NOT USED~~ BANK 8 4 2 1
RAM REGISTER	CHIP SELECT (8 4) | REGISTER SELECT (2 1)	CHARACTER 8 4 2 1

The RAM chip select is addressed by the high order 2 bits of the even register, the RAM register within the selected chip is addressed by the low order 2 bits of the even register and the character within the RAM register is addressed by the 4 bits of the odd register.

Addressing of the I/O port and RAM registers by the even register is tabulated in Table 3.6.3.1. The table covers any one bank of RAM registers. To select other RAM banks refer to the DCL instructions.

TABLE 3.6.3.1

I/O Port and RAM Selection for One Bank by Even Register Contents, as Used with SRC Instruction

CONTENTS OF EVEN REGISTER	I/O PORT SELECTED	RAM # AND RAM REGISTER SELECTED			
		RAM #	REGISTER	RAM DEVICE TYPE	RAM PIN 10 WIRED
0	0	0	0	4002-1	HIGH
1	1	0	1		
2	2	0	2		
3	3	0	3		
4	4	1	0	4002-1	LOW
5	5	1	1		
6	6	1	2		
7	7	1	3		
8	8	2	0	4002-2	HIGH
9	9	2	1		
A	A	2	2		
B	B	2	3		
C	C	3	0	4002-2	LOW
D	D	3	1		
E	E	3	2		
F	F	3	3		

FETCH INDIRECT

FIN

M1				M2			
0	0	1	1	R	R	R	0
8	4	2	1	8	4	2	1

3 P_x0

The 8 bit content of register pair 0 is sent out as an address to the current page of program memory. The 8 bit word at that location is loaded as data into the designated index register pair P_x0. If the FIN occupies the last position of a page, data will be fetched from the next page in sequence. The program counter is not affected.

After the FIN has been executed the next instruction in sequence will be addressed. However, the FIN is a one word instruction, and it requires an additional instruction cycle to retrieve the 8 bits of data for the designated register pair. This extra cycle must be considered when the FIN is used in routines with timing considerations.

FIN uses the even register numbers to designate a pair. The only valid operand codes for P_x0 are 0, 2, 4, 6, 8, A, C, and E. The FIN instruction is useful for retrieving data from look-up or translation tables.

RRR defines one of the eight register pairs P0 through P7. The 0 following RRR is part of the command decoding and distinguishes the FIN from the JIN.

JUMP INDIRECT

JIN

M1				M2			
0	0	1	1	R	R	R	1
8	4	2	1	8	4	2	1

3 P_x1

The 8 bit content of the designated register pair P_x1 is loaded into the low order 8 positions of the program address counter. Program control is transferred to the instruction at that address on the same page. If the JIN occupies the last position of the current page program control transfers to the 8 bit address of the next page in sequence.

The 8 bit content of the register pair is not affected.

JIN uses the odd register numbers to designate the pair P_x. The only valid operand codes for P_x1 are 1, 3, 5, 7, 9, B, D, and F.

RRR defines one of the eight register pairs P0 through P7. The 1 following RRR is part of the command decoding and distinguishes the JIN from the FIN.

JUMP UNCONDITIONAL JUN

M1	M2		
0 1 0 0	A_3	First Word	4 A_3
8 4 2 1	8 4 2 1		
A_2	A_1	Second Word	A_2 A_1
8 4 2 1	8 4 2 1		

Program control is unconditionally transferred to the instruction located at the address A_3, A_2, and A_1. The CPU accomplishes this internally by transferring A_3 from the operand of the instruction register and A_2, A_1 from program memory to the program address counter.

JUMP TO SUBROUTINE JMS

M1	M2		
0 1 0 1	A_3	First Word	5 A_3
8 4 2 1	8 4 2 1		
A_2	A_1	Second Word	A_2 A_1
8 4 2 1	8 4 2 1		

The subroutine address stack is pushed down one level. The program address counter, containing the 12 bit address of the instruction following the second word of the JMS, is transferred to the topmost stack level. Program control is transferred to the instruction located at A_3, A_2, and A_1 from program memory to the program address counter.

First word. In Instruction Register 5 A_3

Second word. From Program Memory A_2 A_1

Program Address Counter P_h P_m P_l

Subroutine Address Stack → (Level 1) [Level 2] {Level 3}

Stack shown fully loaded

INCREMENT REGISTER

INC

M1				M2
0	1	1	0	R_x
8	4	2	1	8 4 2 1

6 R_x

The 4 bit content of the designated register R_x is incremented by 1. If the count causes the register to overflow, the register is set to zero.

The carry bit and accumulator are not affected.

INCREMENT REGISTER STEP IF ZERO

ISZ

M1				M2
0	1	1	1	R_x
8	4	2	1	8 4 2 1

First Word

7 R_x

A_2	A_1
8 4 2 1	8 4 2 1

Second Word

A_2 A_1

The contents of the designated register R_x is incremented by 1. If the result is zero, program control continues in sequence. If the result is not zero, program control is transferred to the instruction located at the 8 bit address A_2, A_1 on the same page. If the ISZ occupies the last two positions of a page, or overlaps the page boundary, program control is transferred to the 8 bit address on the next page in sequence.

The accumulator and carry are not affected.

ISZ is one of the two decision making in structions of the CPU. The other is JCN. ISZ allows a program control decision to be made based on the count of a register.

ADD REGISTER TO ACCUMULATOR

ADD

M1				M2
1	0	0	0	R_x
8	4	2	1	8 4 2 1

8 R_x

The 4 bit content of the designated index register R_x is added to the contents of the accumulator with carry. The result is stored in the accumulator. The carry is set to 1 if a sum greater than 15 was generated, otherwise the carry is set to 0.

The contents of the index register is not affected.

Accumulator in	$a_8\ a_4\ a_2\ a_1$	Augend
Carry in	C_{in}	Carry in
Register	+) $R_8\ R_4\ R_2\ R_1$	Addend
Accumulator out	$a_8\ a_4\ a_2\ a_1$	Sum
Carry out	C_{out}	Carry out

Addition of words longer than 4 bits (multiple precision addition) may be accomplished by starting with the LSD, working on 4 bits at a time until the desired word length has been operated on. It is important not to modify the carry bit between each 4 bits.

SUBTRACT REGISTER FROM ACCUMULATOR — SUB

M1				M2
1	0	0	1	R_X
8	4	2	1	8 4 2 1

9 R_X

The 4 bit content of the designated register R_X is subtracted from the accumulator with borrow. The result is stored in the accumulator. If a borrow is generated, (i.e., R_X > accumulator) the carry bit is set to 0; is a borrow is not generated the carry is set to 1.

The content of the index register R_X is not affected.

The CPU performs the subtraction by adding the complement of the index register plus the complement of the carry to the accumulator.

Accumulator in	$a_8\ a_4\ a_2\ a_1$	Minuend
Carry in	$\overline{C_{in}}$	Borrow in
Register	+) $\overline{R_8}\ \overline{R_4}\ \overline{R_2}\ \overline{R_1}$	Subtrahend
Accumulator out	$a_8\ a_4\ a_2\ a_1$	Result
Carry out	C_{out}	Borrow out

Subtraction of words longer than 4 bits (multiple precision subtraction) can be accomplished by starting with the LSD, working on 4 bits at a time until the desired word length has been operated on. It is required that the carry bit be complemented between each 4 bits for the correct result.

The SUB instruction is useful for performing a compare function. The compare is performed by initially clearing the carry bit and subtracting the 4 bit word R_X to be compared from the accumulator.

The conditions to be tested for comparison results following subtraction are presented below:

COMPARISON	ACCUMULATOR	CARRY	MNEMONIC TEST CONDITION
REG > ACC	$\neq 0$	0	CO
REG = ACC	0	1	AO
REG < ACC	$\neq 0$	1	Al·Cl
REG $\leqq$ ACC	X	1	Cl
REG $\neq$ ACC	$\neq 0$	X	Al

LOAD REGISTER TO ACCUMULATOR LD

M1				M2
1	0	1	0	R_X
8	4	2	1	8 4 2 1

A R_X

The 4 bit content of the designated index register R_X is loaded into the accumulator. The previous contents of the accumulator are lost.

The content of the index register and the carry bit are not affected.

EXCHANGE REGISTER WITH ACCUMULATOR XCH

M1				M2
1	0	1	1	R_X
8	4	2	1	8 4 2 1

B R_X

The 4 bit content of the designated index register R_X is loaded into the accumulator. The prior content of the accumulator is loaded into the designated register R_X.

The carry bit is not affected.

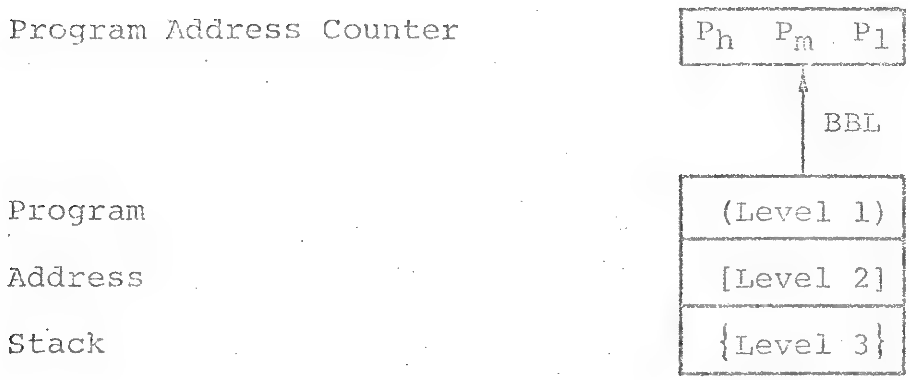
This is the only instruction which allows the accumulator to be loaded into an index register.

BRANCH BACK AND LOAD ACCUMULATOR BBL

M1				M2
1	1	0	0	D_X
8	4	2	1	8 4 2 1

C D_X

BBL is used to return from subroutine to main program. The subroutine address stack is pulled up one level. The top-most address is placed in the program address counter causing program control to be transferred to the sequential instruction following the previous JMS.



Stack shown fully loaded

The 4 bits of data D_x in the operand portion of the instruction are loaded into the accumulator. The previous accumulator data is lost.

The carry bit is not affected.

LOAD DATA TO ACCUMULATOR — LDM

M1				M2			
1	1	0	1	D_1			
8	4	2	1	8	4	2	1

D D_1

The 4 bits of data D_1 stored in the operand field of the instruction word are loaded into the accumulator. The previous contents of the accumulator are lost.

The carry bit is not affected.

WRITE ACCUMULATOR INTO RAM CHARACTER — WRM

M1				M2			
1	1	1	0	0	0	0	0
8	4	2	1	8	4	2	1

E0

The accumulator content is written into the RAM main memory character location previously selected by an SRC instruction.

The accumulator and carry are not affected.

WRITE MEMORY PORT — WMP

M1				M2			
1	1	1	0	0	0	0	1
8	4	2	1	8	4	2	1

E1

The content of the accumulator is transferred to the RAM output port previously selected by an SRC instruction. The data is available on the output pins until a new WMP is executed on the same RAM chip.

The accumulator and carry are not affected.

WRITE ROM PORT — WRR

M1				M2			
1	1	1	0	0	0	1	0
8	4	2	1	8	4	2	1

E2

The content of the accumulator is transferred to the output port previously selected by an SRC instruction. The data is available on the output pins until a new WRR is executed on the same port.

The accumulator and carry are not affected.

WRITE TO PROGRAM MEMORY — WPM

M1				M2			
1	1	1	0	0	0	1	1
8	4	2	1	8	4	2	1

E3

This instruction is used to write data into RAM program memory 4 bits at a time. The WPM instruction must be executed twice for each 8 bit RAM program memory location.

Program memory page select lines are forced to 1111. The previous SRC address is sent out on the program memory address bus and the accumulator contents becomes available as 4 bits of data on the I/O bus. Two control lines from the CPU interface circuitry control writing into the RAM.

WRITE INTO RAM STATUS CHARACTER 0 — WRO

M1				M2			
1	1	1	0	0	1	0	0
8	4	2	1	8	4	2	1

E4

The content of the accumulator is written into the RAM status character 0 previously selected by an SRC instruction.

The accumulator and carry are not affected.

WRITE INTO RAM STATUS CHARACTER 1 — WR1

M1				M2			
1	1	1	0	0	1	0	1
8	4	2	1	8	4	2	1

E5

The content of the accumulator is written into the RAM status character 1 previously selected by an SRC instruction.

The accumulator and carry are not affected.

WRITE INTO RAM STATUS CHARACTER 2 — WR2

M1				M2			
1	1	1	0	0	1	1	0
8	4	2	1	8	4	2	1

E6

The content of the accumulator is written into the RAM status character 2 previously selected by an SRC instruction.

The accumulator and carry are not affected.

WRITE INTO RAM STATUS CHARACTER 3 — WR3

M1				M2			
1	1	1	0	0	1	1	1
8	4	2	1	8	4	2	1

E7

The content of the accumulator is written into the RAM status character 3 previously selected by an SRC instruction.

The accumulator and carry are not affected.

SUBTRACT FROM MEMORY WITH BORROW

SBM

M1				M2			
1	1	1	0	1	0	0	0
8	4	2	1	8	4	2	1

E8

The content of the RAM character previously selected by an SRC instruction is subtracted from the accumulator with borrow.

The RAM character is unaffected.

READ RAM CHARACTER

RDM

M1				M2			
1	1	1	0	1	0	0	1
8	4	2	1	8	4	2	1

E9

The content of the RAM character is transferred to the accumulator.

The carry is not affected. The 4 bit data in memory is unaffected.

READ ROM PORT

RDR

M1				M2			
1	1	1	0	1	0	1	0
8	4	2	1	8	4	2	1

EA

The data present at the input lines of the port previously selected by an SRC instruction is transferred to the accumulator.

The carry is not affected.

ADD FROM MEMORY WITH CARRY

ADM

M1				M2			
1	1	1	0	1	0	1	1
8	4	2	1	8	4	2	1

EB

The content of the RAM character previously selected by an SRC instruction is added to the accumulator with carry.

The RAM character is not affected.

READ RAM STATUS CHARACTER 0

RDO

M1				M2			
1	1	1	0	1	1	0	0
8	4	2	1	8	4	2	1

EC

The 4 bits of status character 0 of the RAM register previously selected by an SRC instruction are transferred to the accumulator.

The carry and the status character are not affected.

READ RAM STATUS CHARACTER 1 RD1

M1				M2			
1	1	1	0	1	1	0	1
8	4	2	1	8	4	2	1

ED

The 4 bits of status character 1 of the RAM register previously selected by an SRC instruction are transferred to the accumulator.

The carry and the status character are not affected.

READ RAM STATUS CHARACTER 2 RD2

M1				M2			
1	1	1	0	1	1	1	0
8	4	2	1	8	4	2	1

EE

The 4 bits of status character 2 of the RAM register previously selected by an SRC instruction are transferred to the accumulator.

The carry and the status character are not affected.

READ RAM STATUS CHARACTER 3 RD3

M1				M2			
1	1	1	0	1	1	1	1
8	4	2	1	8	4	2	1

EF

The 4 bits of status character 3 of the RAM register previously selected by an SRC instruction are transferred to the accumulator.

The carry and the status character are not affected.

CLEAR BOTH CLB

M1				M2			
1	1	1	1	0	0	0	0
8	4	2	1	8	4	2	1

F0

Set accumulator and carry to 0.

CLEAR CARRY CLC

M1				M2			
1	1	1	1	0	0	0	1
8	4	2	1	8	4	2	1

F1

Set carry to 0.

The accumulator is not affected.

INCREMENT ACCUMULATOR IAC

M1				M2			
1	1	1	1	0	0	1	0
8	4	2	1	8	4	2	1

F2

The content of the accumulator is incremented by 1. No overflow sets the carry to 0; overflow sets the carry to a 1.

COMPLEMENT CARRY CMC

M1				M2			
1	1	1	1	0	0	1	1
8	4	2	1	8	4	2	1

F3

The carry content is complemented.

The accumulator is not affected.

COMPLEMENT ACCUMULATOR CMA

M1				M2			
1	1	1	1	0	1	0	0
8	4	2	1	8	4	2	1

F4

The content of the accumulator is complemented.

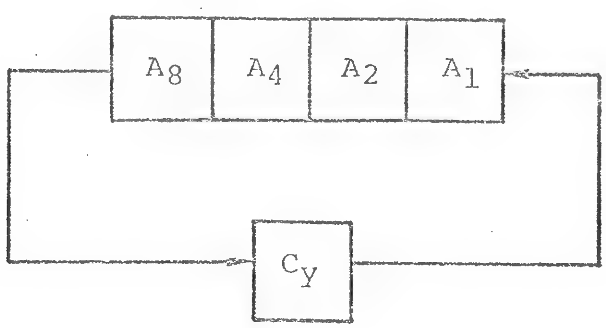
The carry is not affected.

ROTATE LEFT RAL

M1				M2			
1	1	1	1	0	1	0	1
8	4	2	1	8	4	2	1

F5

The content of the accumulator and carry are rotated left one bit position.

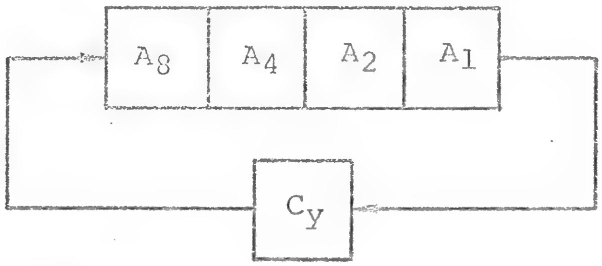


ROTATE RIGHT — RAR

M1				M2			
1	1	1	1	0	1	1	0
8	4	2	1	8	4	2	1

F6

The content of the accumulator and carry are rotated right 1 bit position.



TRANSMIT CARRY AND CLEAR — TCC

M1				M2			
1	1	1	1	0	1	1	1
8	4	2	1	8	4	2	1

F7

The accumulator is cleared. The least significant position of the accumulator is set to the value of the carry. The carry is set to 0. This instruction is used for decimal arithmetic.

DECREMENT ACCUMULATOR — DAC

M1				M2			
1	1	1	1	1	0	0	0
8	4	2	1	8	4	2	1

F8

Decrementing when the accumulator equals zero sets the carry to 0. Decrementing when the accumulator is not zero sets the carry to 1. The initial value of the carry bit does not affect the content of the accumulator.

TRANSFER CARRY SUBTRACT — TCS

M1				M2			
1	1	1	1	1	0	0	1
8	4	2	1	8	4	2	1

F9

The accumulator is set to 9 if the carry is 0. The accumulator is set to 10 if the carry is a 1. The carry is set to 0. This instruction is used for decimal arithmetic.

SET CARRY

STC

M1				M2			
1	1	1	1	1	0	1	0
8	4	2	1	8	4	2	1

FA

Set carry to a 1.

The accumulator is not affected.

DECIMAL ADJUST ACCUMULATOR

DAA

M1				M2			
1	1	1	1	1	0	1	1
8	4	2	1	8	4	2	1

FB

The accumulator is incremented by 6 if either the carry is 1 or if the accumulator content is greater than 9. The carry is set to a 1 if the result generates a carry, otherwise it is unaffected. This instruction is used for decimal arithmetic.

KEYBOARD PROCESS

KBP

M1				M2			
1	1	1	1	1	1	0	0
8	4	2	1	8	4	2	1

FC

A code conversion is performed on the accumulator content, from 1 out of n to binary code. If the accumulator content has more than 1 bit on, the accumulator will be set to 15 (to indicate error). The carry is not affected. The conversion table is shown below:

(ACC) before KBP			(ACC) after KBP	
Hex	Binary		Binary	Hex
0	0 0 0 0	→	0 0 0 0	0
1	0 0 0 1	→	0 0 0 1	1
2	0 0 1 0	→	0 0 1 0	2
4	0 1 0 0	→	0 0 1 1	3
8	1 0 0 0	→	0 1 0 0	4
3	0 0 1 1	→	1 1 1 1	F
5	0 1 0 1	→	1 1 1 1	F
6	0 1 1 0	→	1 1 1 1	F
7	0 1 1 1	→	1 1 1 1	F
9	1 0 0 1	→	1 1 1 1	F
A	1 0 1 0	→	1 1 1 1	F
B	1 0 1 1	→	1 1 1 1	F
C	1 1 0 0	→	1 1 1 1	F
D	1 1 0 1	→	1 1 1 1	F
E	1 1 1 0	→	1 1 1 1	F
F	1 1 1 1	→	1 1 1 1	F

FD

M1				M2			
1	1	1	1	1	1	0	1
8	4	2	1	8	4	2	1

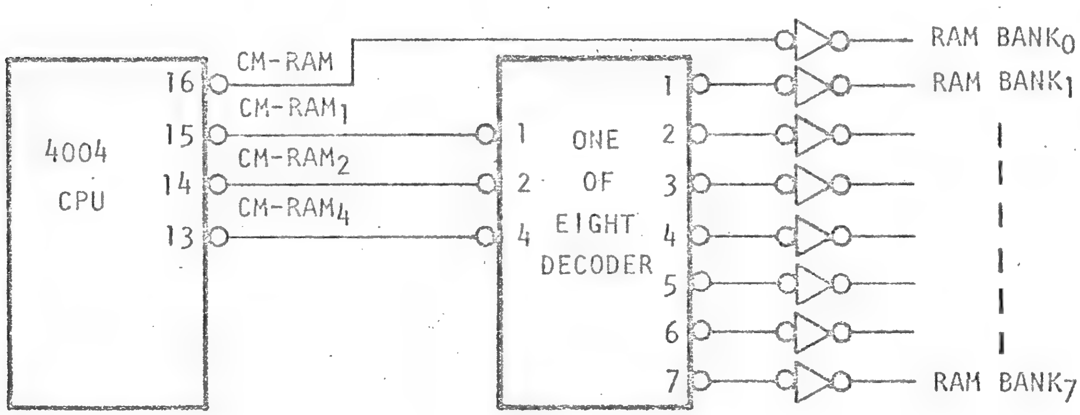
The content of the three least significant accumulator bits is transferred to the CM-RAM output lines on the CPU.

This instruction provides RAM bank selection when multiple RAM banks are used. When the CPU is reset, RAM Bank zero is automatically selected. DCL remains latched until it is changed or reset.

The selection is made according to the following truth table:

(ACC) 8421	CM-RAM_i Enabled	Bank No.
X000	CM-RAM	Bank 0
X001	CM-RAM_1	Bank 1
X010	CM-RAM_2	Bank 2
X100	CM-RAM_4	Bank 3
X011	CM-RAM_1, CM-RAM_2	Bank 4
X101	CM-RAM_1, CM-RAM_4	Bank 5
X110	CM-RAM_2, CM-RAM_4	Bank 6
X111	CM-RAM_1, CM-RAM_2, CM-RAM_4	Bank 7

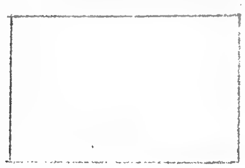
A low power TTL one-of-eight decoder may be tied to the CM-RAM_1, CM-RAM_2, and CM-RAM_4 lines to expand the number of RAM banks to 8. The command lines must be buffered for MOS compatibility.



3.7 FLOWCHART SYMBOLS AND GUIDELINES

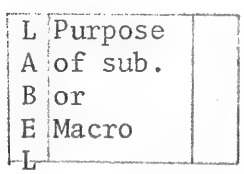
The purpose of a flowchart is to improve communication between analyst and programmer relative to the analysis of a certain processing problem. Flowcharting is a technique in which symbols represent both the sequence of operation and the flow of data. Normal direction of flow is from left to right and top to bottom. All flowcharts should comply with USA Standard Flowchart Symbols and Their Usage in Information Processing (USAS x 3.5 - 1968).

1. Process Symbol



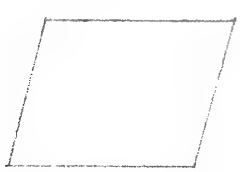
This symbol is used for the process of executing a defined operation of group of operations.

2. Predefined Process (Subroutine, Macro)



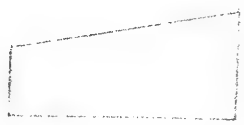
The left hand division marked "Label" contains the calling label or name of the subroutine or macro. The middle section is used to briefly define the purpose of the subroutine or macro.

3. Input/Output Symbol (I/O)



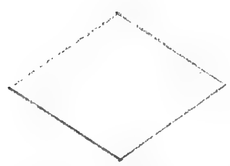
This symbol is used for all I/O processing except that processing which is manual.

4. Manual Input Symbol



This symbol represents an I/O function in which the information is entered manually at the time of processing, (run time), by means of online keyboards, switch settings, and pushbuttons.

5. Decision Symbol

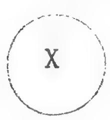


The decision symbol represents a decision or switching type operation that determines which of a number of alternative paths is to be followed.

6. Start, Stop, Pause, Halt, Exit, Return Symbol

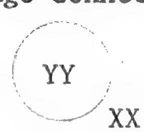
This symbol represents a terminal point in a system at which the label is the exit or entry point.

7. On Page Connector



X indicates an alpha character (A, B, C, etc.) indicating a connection to another on page connector with the same character. See note.*

8. Off Page Connector



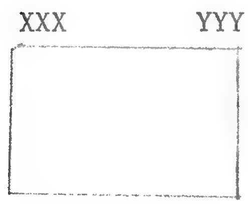
XX is used as a cross reference and indicates the page number of flow destination or origination.

YY is the connection number. The connector number is identical on both the originating and receiving page. See note.*

9. Annotation Symbol

This symbol represents the addition of descriptive comments or explanatory notes as clarification.

10. Symbol Indentification



XXX is a cross reference to the code and is in multiples of 10 to allow insertion of additional numbers.

YYY is the label used, if any, in the code.

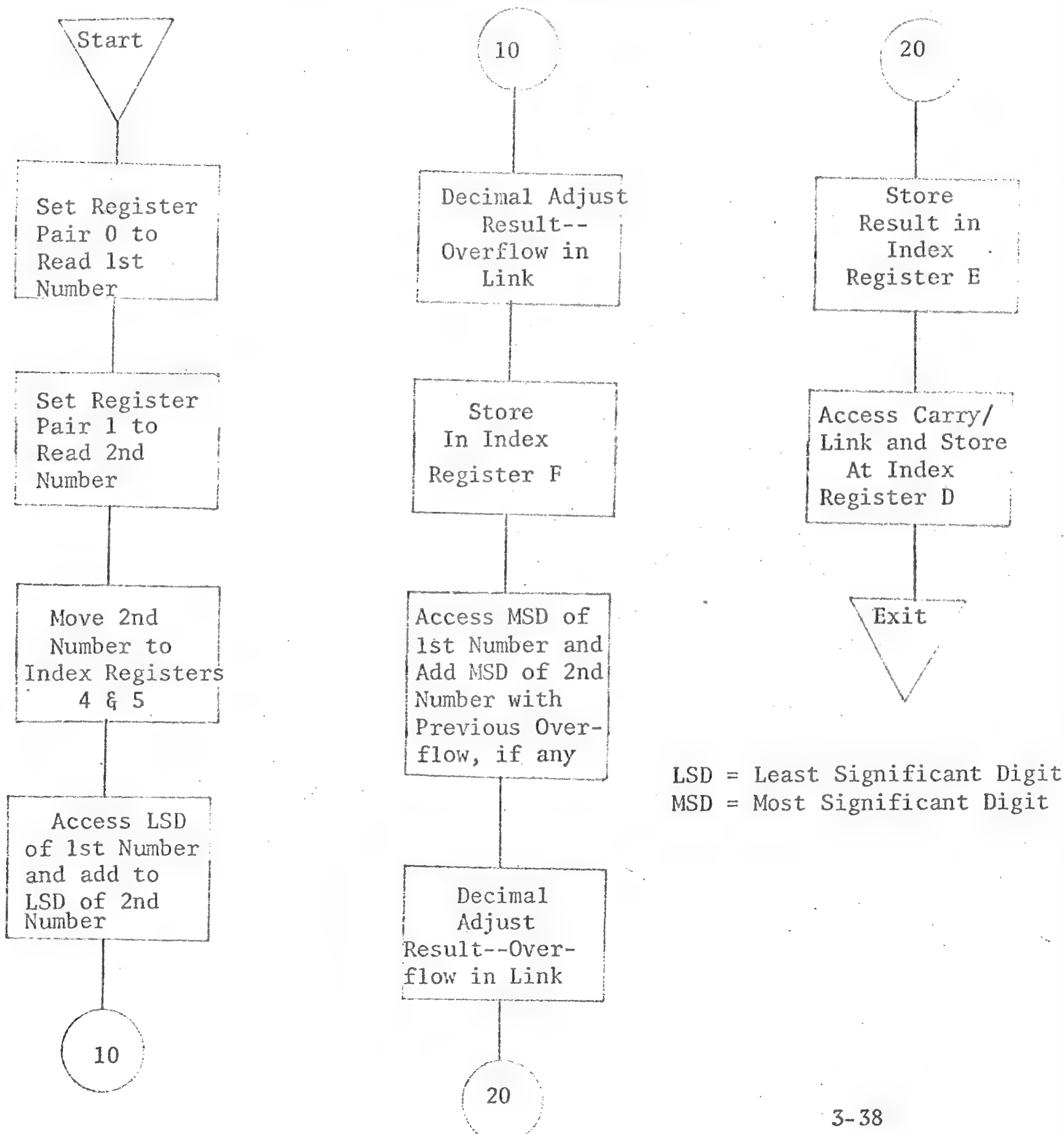
* NOTE: The left and right columns on the flowchart form contain only off page connectors, on page connectors, or terminal label (START, STOP, etc.)

3.8 Programming Examples:

3.8.1 Addition

Problem — Add two 2 digit BCD numbers and store result

Solution — Assume each 2 digit number is stored in RAM one number in RAM 0, Register 0, Character 0, 1; the second number is in RAM 1, Register 2, Character A, B. The result including carry, will be stored in Index Registers D, E, F



Add 2, 2 digit BCD numbers and store result, with carry, in index registers D, E, F

ADDRESS	CODE	LABEL	OPR	OPA1	OPA2	COMMENTS
10	20	Start	FIM	0	0	1st number in RAM 0, register 0,
11	00					0 and 1 (LSD First)
12	22		FIM	1	6A	2nd number in RAM 1, register 1,
13	6A					A and B (LSD First)
14	23		SRC	1		
15	E9		RDM			Read MSD
16	B5		XCH	5		Store in Index register 4
17	63		INC	3		
18	23		SRC	1		
19	E9		RDM			Read LSD
1A	B4		XCH	4		Store in Index register 5
1B	F1		CLC			Clear carry/link bit
1C	21		SRC	0		
1D	E9		RDM			Read LSD, 1st number
1E	84		ADD	4		Add LSD, 2nd number
1F	FB		DAA			Decimal adjust
20	BF		XCH	F		Store at Index register F
21	61		INC	1		
22	21		SRC	0		
23	E9		RDM			Read MSD, 1st number
24	85		ADD	5		Add MSD, 2nd number
25	FB		DAA			with carry & decimal adjust
26	BE		XCH	E		store at index register E
27	F7		TCC			carry/link bit at accumulator
28	BD		XCH	D		

3.8.2 Paper Tape Punch

Generating a punched paper tape is a good test for both the microcomputer and the paper punch. The purpose of the program is to generate four alternative holes per line on an 8 level paper tape. The punched paper tape should look like:

```
|0 1 0 1 0 1 0 1|  <------ Paper Tape
|               |          "1" a punch
|1 0 1 0 1 0 1 0|          "0" a blank
```

Figure: TAPE

GEN TAPE WITH 55 AA 55 AA ON PUNCH

ADDRESS	MACHINE CODE	LABEL	MNEMONIC CODE OPR	OPA1	OPA2	COMMENTS
10	26		FIM	3	5 5	Register Pair 3 - 55
11	55					
12	11	L1	JCN	TZ	L1	Punch sync
13	12					
14	22		FIM	1	1 0	Data address
15	10					
16	23		SRC	1		Connect
17	A7		LD	7		Fetch lower 4 holes
18	E2		WRR			Output
19	62		INC	2		Increment connect address
1A	23		SRC	1		Connect
1B	A6		LD	6		Fetch upper 4 holes
1C	E2		WRR			Output
1D	62		INC	2		Increment connect address
1E	23		SRC	1		Connect
1F	D1		LDM	1		Punch Strobe
20	E2		WRR			On
21	D0		LDM	0		Clear accumulator
22	E2		WRR			Punch strobe off
23	F1		CLC			
24	A7		LD	7		
25	F5		RAL			Rotate left
26	B7		XCH	7		Put back
27	A6		LD	6		
28	F5		RAL			Rotate left
29	1A		JCN	C0	L1	Jump if carry - 0
2A	12					
2B	67		INC	7		Inc 7
2C	40		JUN	L1		Loop
2D	12					

Explanation of the Program:

Close observation of the figure: TAPE reveals that first line is 55_{16} (0 1 0 1 0 1 0 1) and the second line is AA_{16} (1 0 1 0 1 0 1 0).

Register pair 3 (i.e. Index register 6 and index register 7) is used to store 5_{16} in them as shown. (FIM 3 55)

Register 6 and 7

0 1 0 1	0 1 0 1
4 bits	4 bits

Then the microcomputer is electrically connected to the paper tape punch. (SRC 3). After this connection is made, 5_{16} in register 7 and then 5_{16} in register 6 is output. At this stage the paper tape looks like:

0 1 0 1 0 1 0 1	← Paper Tape "1" a punch "0" a blank

Now four bits of register 7 are brought to the accumulator. (LD 7)

Accumulator

0 1 0 1
4 bits

This is shifted one bit to the left. (RAL)

Accumulator
1 0 1 0

The new value, which is A_{16}, is stored back into register 7. (XCH7)

Register 7
1 0 1 0

Above four steps are repeated to achieve A_{16} in register 6.

Register 6
1 0 1 0

Register 7 and 6 are then output to the paper tape punch. The tape now looks like:

0 1 0 1 0	1 0 1
1 0 1 0 1	0 1 0

3.8.3 Output of Data to Typesetting Punch

This program is almost the same as example 3.5.2. The only difference is, this program checks the status of Paper Tape Punch before sending information to it.

The JCN TZ, BBL0 is the conditional jump if the paper tape punch synch bit is high (logical 1) on the test signal line.

JCN AN PUNCH - If the accumulator is non zero, (AN) the program branches to the address designated by the label punch.

Output of data to typesetting punch
(data previously stored in buffer)

ADDRESS	CODE	LABEL	OPR	OPA1	OPA2	COMMENTS
20	11	Test	JCN	TZ	BBL0	Test for Punch Synch
21	48					
22	D1		LDM	1		
23	B6		XCH	6		
24	27		SRC	3		
25	AF		LD	F		Test whether punch buffer-empty
26	F1		CLC			Reg pair 7 points to the word
27	9D		SUB	D		before the punch buffer. Reg pr.
28	1C		JCN	AN	PUNCH	6 points to the last word in
29	30					punch buffer.
2A	E2		WRR			The two reg pairs are equal, the
2B	AE		LD	E		output buffer is empty.
2C	F1		CLC			
2D	9C		SUB	C		
2E	14		JCN	AZ	BBL15	If buffer is not empty punch
2F	BC					character.
30	7F	Punch	ISZ	F	GOOD	Increment reg pair 7
31	36					
32	7E		ISZ	E	GOOD	
33	36					
34	2E		FIM	7	4; 0	Reset reg pair 7 to beginning
35	40					of buffer
36	2F	Good	SRC	7		Load address of buffer
37	E9		RDM			Read upper 2 bits from buffer
38	27		SRC	3		Load punch address
39	E2		WRR			Send upper 2 bits from buffer
3A	6F		INC	F		Increment buffer address
3B	66		INC	6		Increment punch address
3C	2F		SRC	7		Load buffer address
3D	E9		RDM			Read lower 4 bits from buffer
3E	27		SRC	3		Load punch address
3F	E2		WRR			Send lower 4 bits to punch
40	66		INC	6		Set control word address
41	27		SRC	3		Load control word address
42	EC		RDO			Set punch strobe bit
43	F2		IAC			
44	E2		WRR			Send punch strobe bit and keep
45	F8		DAC			reader strobe on decrement
46	E2		WRR			punch strobe bit
47	C1		BBL	1		Return to calling program
48	C0		BBL	0		Exit

3.8.4. Pulse Generator

Generating a pulse for a given time is vital to many electronic components for operation. A program is written to output a pulse for every 100 microseconds. This program consists of one main program and two subprograms.

The program initially outputs a zero on the line (1 bit) and delays for 100 milleseconds. Then the program outputs a one on the line for 100 milleseconds. This sequence is repeated indefinitely while the program is running.

OUTPUT ON EACH LINE A PULSE 100 ms ON, ON ADDRESS 00 AND 10

ADDRESS	CODE	LABEL	OPR	OPA1	OPA2	COMMENTS
10	20	START	FIM	0	0 < 0	Select address to reg. pair 0
11	00					
12	21		SRC	0		Connect I/O
13	50		JMS	INOUT		Call I/O
14	1B					
15	60		INC	0		Increment select address
16	21		SRC	0		Connect I/O
17	50		JMS	INOUT		Call I/O
18	1B					
19	40		JUN	START		Loop
1A	10					
1B	FA	INOUT	STC			
1C	D0		LDM	0		
1D	F5	L1	RAL			
1E	E2		WRR			Output line on
1F	50		JMS	DELAY		Delay 100 milleseconds
20	25					
21	B7		XCH	1		Restore accumulator
22	1A		JCN	C_y = 1,	1	Jump if CY = 0
23	1D					
24	C0		BBL	0		Return to caller
25	22	DELAY	FIM	1	0 < 0	
26	00					
27	24		FIM	2;	2 < 0	
28	20					
29	72	L5	ISZ	2;	L5	
2A	29					
2B	73		ISZ	3;	L5	
2C	29					
2D	74		ISZ	4;	L5	
2E	29					
2F	B7		XCH	7		Save accumulator
30	C0		BBL	0		Return

3.8.5 Multiply Routine

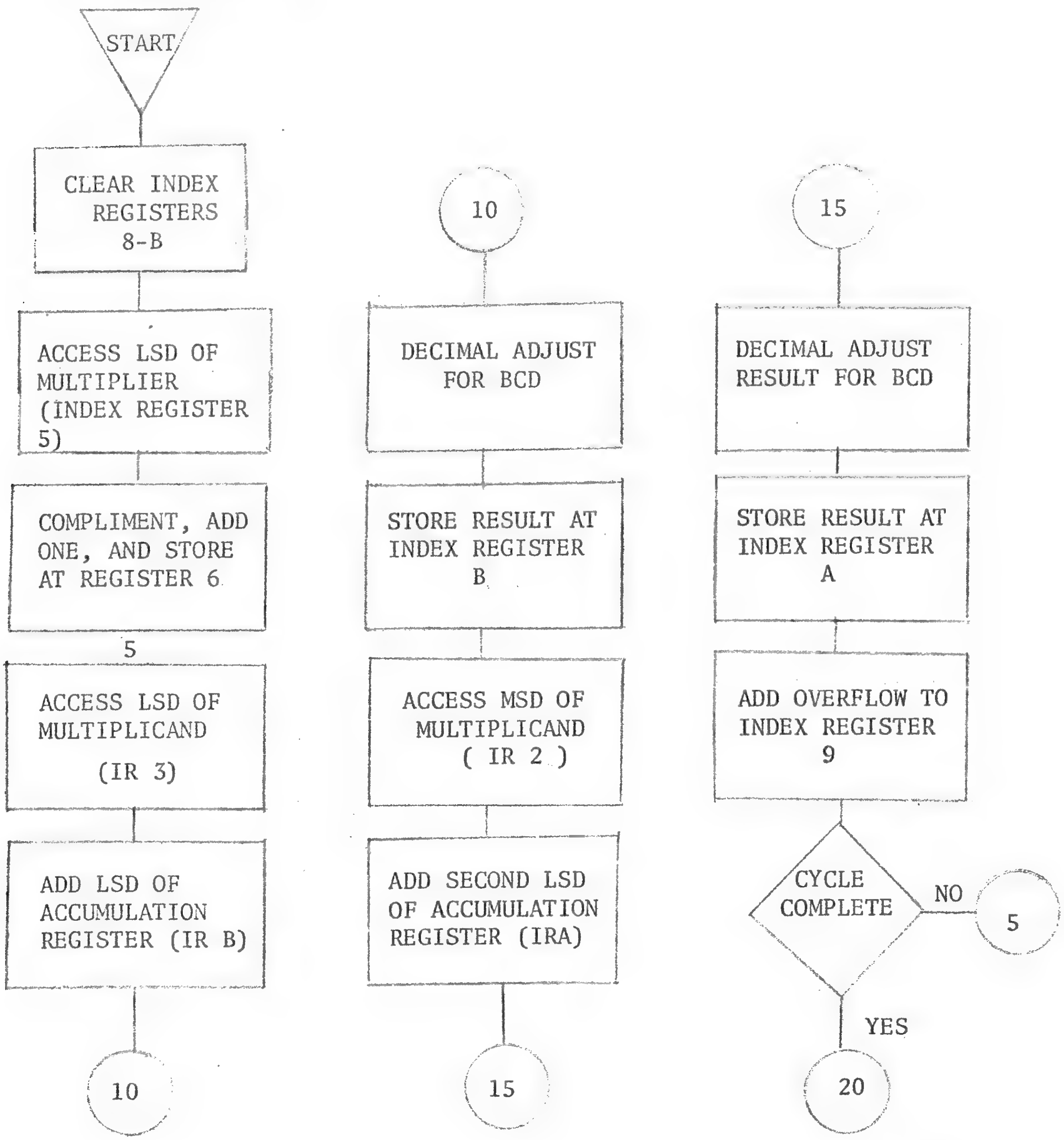
Problem--multiply 2, 2 digit BCD numbers and store, the result (maximum of 4 digits).

Solution--Assume that the 2 numbers are residing in 4 index registers (2, 3, 4, 5) Successive additions will provide the required result with Index Register 6 the counter for controlling the successive additions. Index Registers 8, 9, A, B will hold the final result of the multiplication (Accumulation Registers).

LSD = Least Significant Disit.

MSD = Most Significant Disit.

IR = Index Register

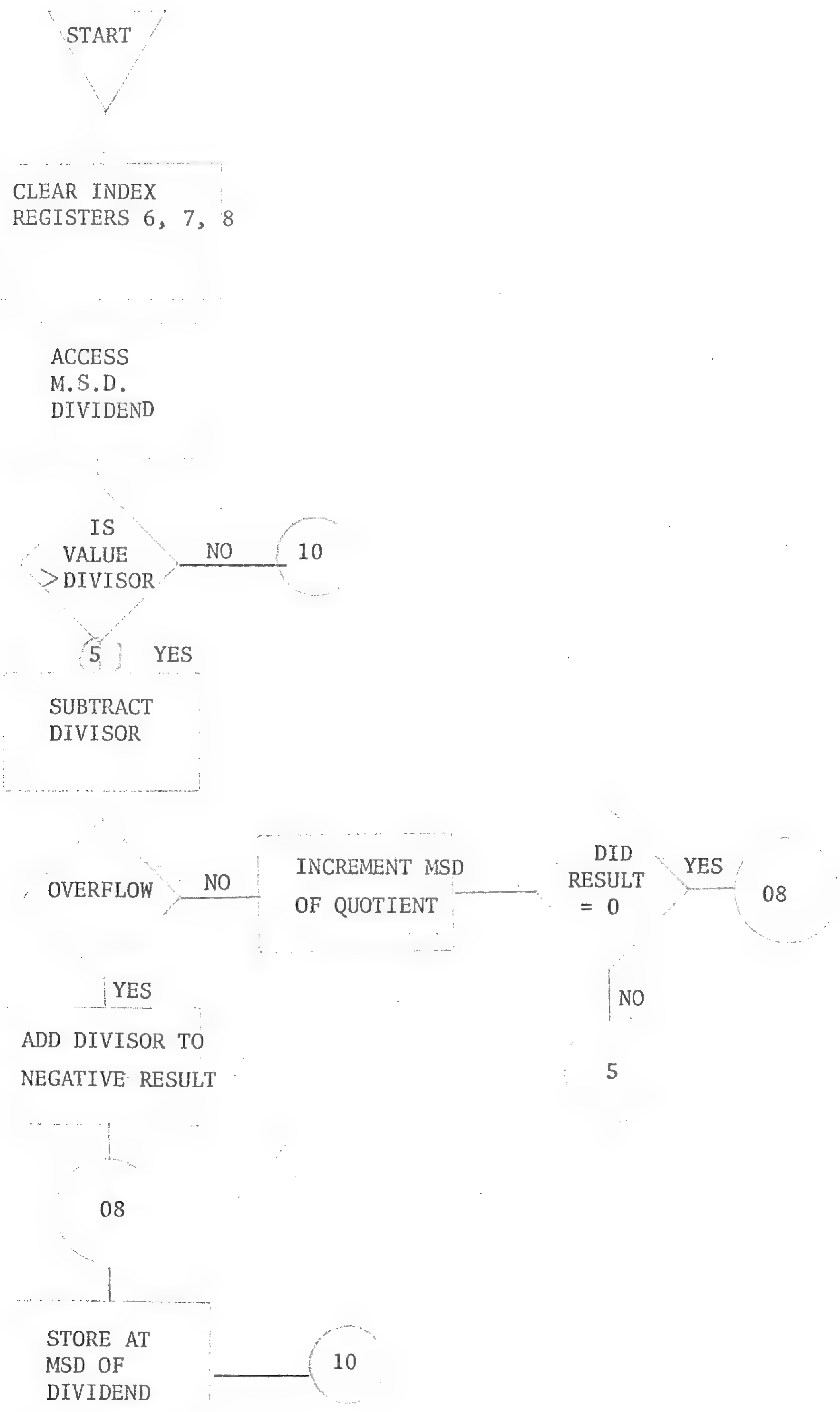


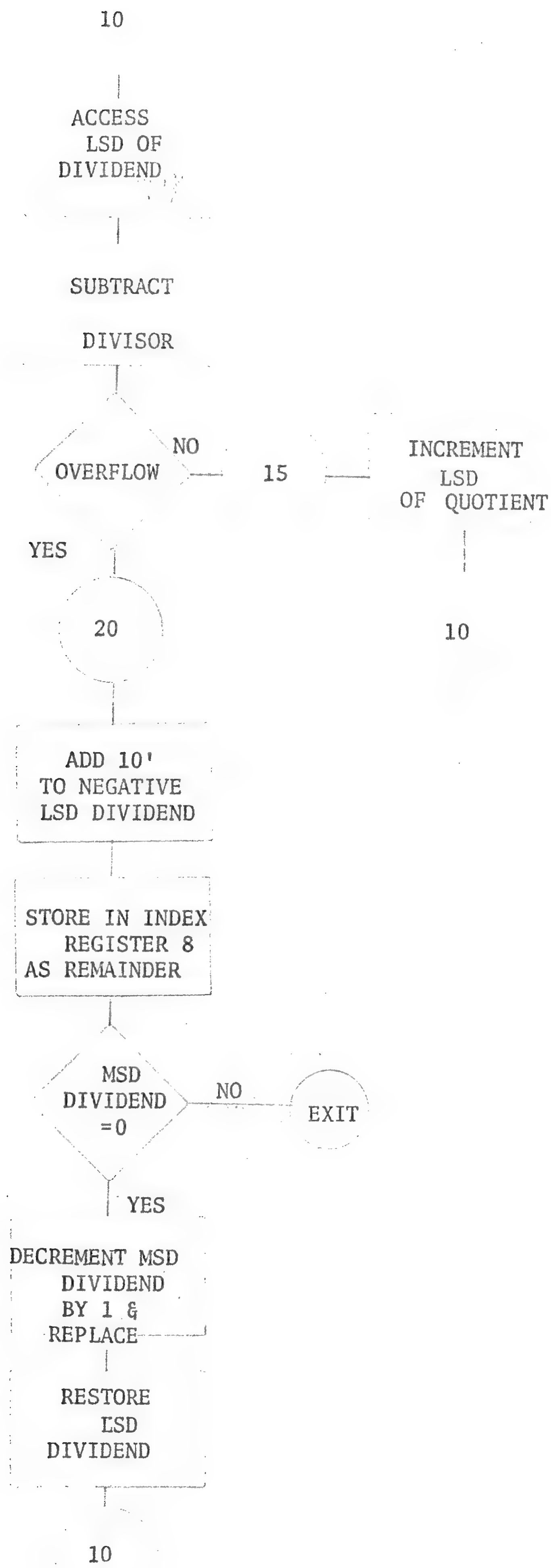
ADDRESS	CODE	LABEL	OPR	OPA1	OPA2	COMMENTS
10	28	START	FIM	4	00	Clear IR
1	00					
2	2A		FIM	5	00	8-B
3	00					
4	A5		LD	5		LSD Multiplier,
5	F4		CMA			Compliment,
6	F2		IAC			add one,
7	B6		XCH	6		Store
8	FO	MULT 05	CLB			
9	A3		LD	3		LSD Multiplicand
A	8B		ADD	B		Add LSD
B	FB	MULT 10	DAA			Accumulation,
C	BB		XCH	B		Decimal Adjust, Store
D	AZ		LD	2		MSD Multiplicand
E	8A		ADD	A		Add 2nd LSD of
F	FB	MULT 15	DAA			Accumulation,
20	BA		XCH	A		Decimal Adjust, Store
1	DO		LDM	0		
2	89		ADD	9		Overflow to
3	B9		XCH	9		IR 3
4	76		ISZ	6	MULTO5	if complete
5	18					
6	A4	MULT 20	LD	4		Go on--
7	F4		CMA			Loop control
8	F2		IAC			(MSD of multiplier)
9	B6		XCH	6		
A	FO	MULT 25	CLB			
B	A3		LD	3		LSD of multiplicand
C	8A		ADD	A		Plus 2nd LSD of
D	FB		DAA			Accumulation,
E	BA		XCH	A		Adjust & store
F	A2		LD	2		MSD Multiplicand
30	89	MULT30	ADD	9		Plus 3rd LSD of
1	FB		DAA			Accumulation
2	B9		XCH	9		Adjust & Store
3	DO		LDM	0		Store
4	88		ADD	8		Overflow
5	B8		XCH	8		
6	76		ISZ	6	MULT25	
7	2A					

3.8.6 DIVISION

Divide a two digit BCD number by a 1 digit BCD number. The divisor is prestored at Index Register 4, the dividend is prestored at Index Registers 2 and 3 (I.R. 3 = LSD). The Quotient is stored at Index Registers 6 and 7 with a truncated remainder at Index Register 8.

Division will be preformed by successive subtractions.


START
CLEAR INDEX REGISTERS 6, 7, 8
ACCESS M.S.D. DIVIDEND
IS VALUE > DIVISOR
NO
10
5
YES
SUBTRACT DIVISOR
OVERFLOW
NO
INCREMENT MSD OF QUOTIENT
DID RESULT = 0
YES
08
NO
5
YES
ADD DIVISOR TO NEGATIVE RESULT
08
STORE AT MSD OF DIVIDEND
10


10
ACCESS LSD OF DIVIDEND
SUBTRACT DIVISOR
OVERFLOW
NO
15
INCREMENT LSD OF QUOTIENT
10
YES
20
ADD 10' TO NEGATIVE LSD DIVIDEND
STORE IN INDEX REGISTER 8 AS REMAINDER
MSD DIVIDEND =0
NO
EXIT
YES
DECREMENT MSD DIVIDEND BY 1 & REPLACE
RESTORE LSD DIVIDEND
10

DIVISON EXAMPLE
IR = Index Register
MSD = Most Significant Disit.
LSD = Least Significant Disit.

ADDRESS	CODE	LABEL	OPR	OPA1	OPA2	COMMENTS
10	26	START	FIM	3	0, 0	Clear IR 6, 7
11	00					
12	28		FIM	4	0, A	Clear IR 8, 5
13	0A					Set IR 9 = A
14	FO		CLB			
15	A2	Div 05	LD	2		MSD OF Dividend
16	94		SUB	4		Subtract Divisor
17	1A		JCN	CY=0	DIV 07	Overflow?
18	1F					Yes, got to Div. 07
19	66		INC	6		Increment MSD Quotient
1A	14		JCN	ACC=0	DIV 08	Result = 0
1B	20					Yes, Go to Div. 08
1C	B2		XCH	2		Restore
1D	40		JUN		DIV 05	
1E	15					
1F	84	Div 07	ADD	4		Add Divisor to - result
20	B2	Div 08	XCH	2		
21	FO		CLB			
22	A2	Div 10	LD	3		LSD Dividend
23	94		SUB	4		Subtract Divisor
24	1A		JCN	CY=0	Div 20	Overflow
25	29					
26	B3		XCH	3		Replace
27	67	Div 15	INC	7		Increment LSD Quotient
28	40		JUN	DIV 10		Cycle Again
29	22					
2A	89	Div 20	ADD	9		ADD '10', Store
2B	B8		XCH	8		as remainder
2C	A2		LD	2		
2D	14		JCN	ACC=0	Exit	IF = 0, exit
2E	35					
2F	F8		DAC			Borrow from MSD
30	B2		XCH			
31	A8		LD	8		
32	B3		XCH	3		
33	40		JUN	Div 15		Cycle
34	27					
35		Exit				

Chapter 7

7.0 Introduction

7.1 Machine Control
- 7.1.1 Stepper Motor
- 7.1.2 Hydraulic System
- 7.1.3 D.C. Servo Motor
- 7.1.4 Input/Output Equipment
  - 7.1.4.1 Paper Tape Reader
  - 7.1.4.2 Paper Tape Punch
  - 7.1.4.3 Magnetic Tape Cartridge
  - 7.1.4.4 Teleprinter
  - 7.1.4.5 Multiplexer Driver
  - 7.1.4.6 Diode Board

7.2 Remote Terminal/Data Entry and Preparation

7.3 Inventory Control and Analysis

7.4 Point of Entry Pricing

7.5 Conveyor Control System

7.6 O E M

CHAPTER 7

7.0 Introduction

The CPU of the Comstar 4 has a powerful and versatile instruction set which allows the system to perform a wide variety of arithmetic, control and decision functions. The PROM, with stored microprograms, provides users the capability of designing custom computers with standard components. The Comstar 4 is useable in a wide variety of areas from process control to the field of data entry. Examples of the Comstar is use are:

7.1 MACHINE CONTROL

Background

In the late 60's computers became powerful tools as controllers. However, applying computers effectively to every area was not easily accomplished. One particular difficult area was in the numerical control of machine tools.

7.1.1 When computerized numberical control began, it was economical to use hard-wired controllers for each task. Since the prices of general purpose computers have dropped, they became more feasible to use. The major benefit of the general purpose computer to control a NC-system is that it permits new features to be easily implemented by changing the software.

General purpose computer applications have two general areas.

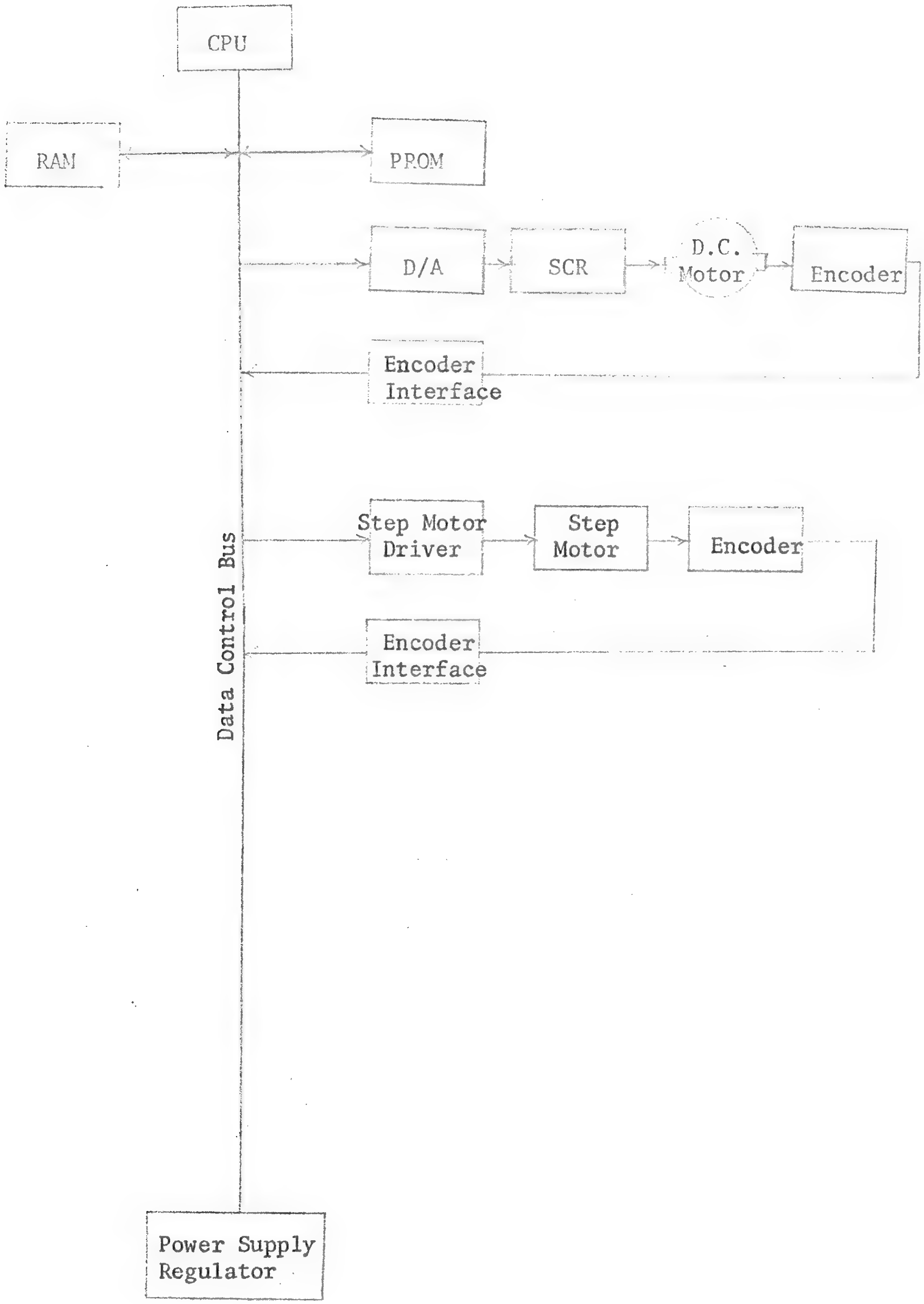
1. Direct Numerical Control - where a large computer works as a controller for more than five machines.

2. Computer Numerical Control - where a minicomputer controls one or two machines.

Among DNC and CNC, CNC is preferred since it's very flexible. Large computers are expensive to own and if the computer is down all machines it controls will be down. One the otherhand, minicomputers are less expensive and if the minicomputer is down, only one or two machines that are connected to that computer will be affected.

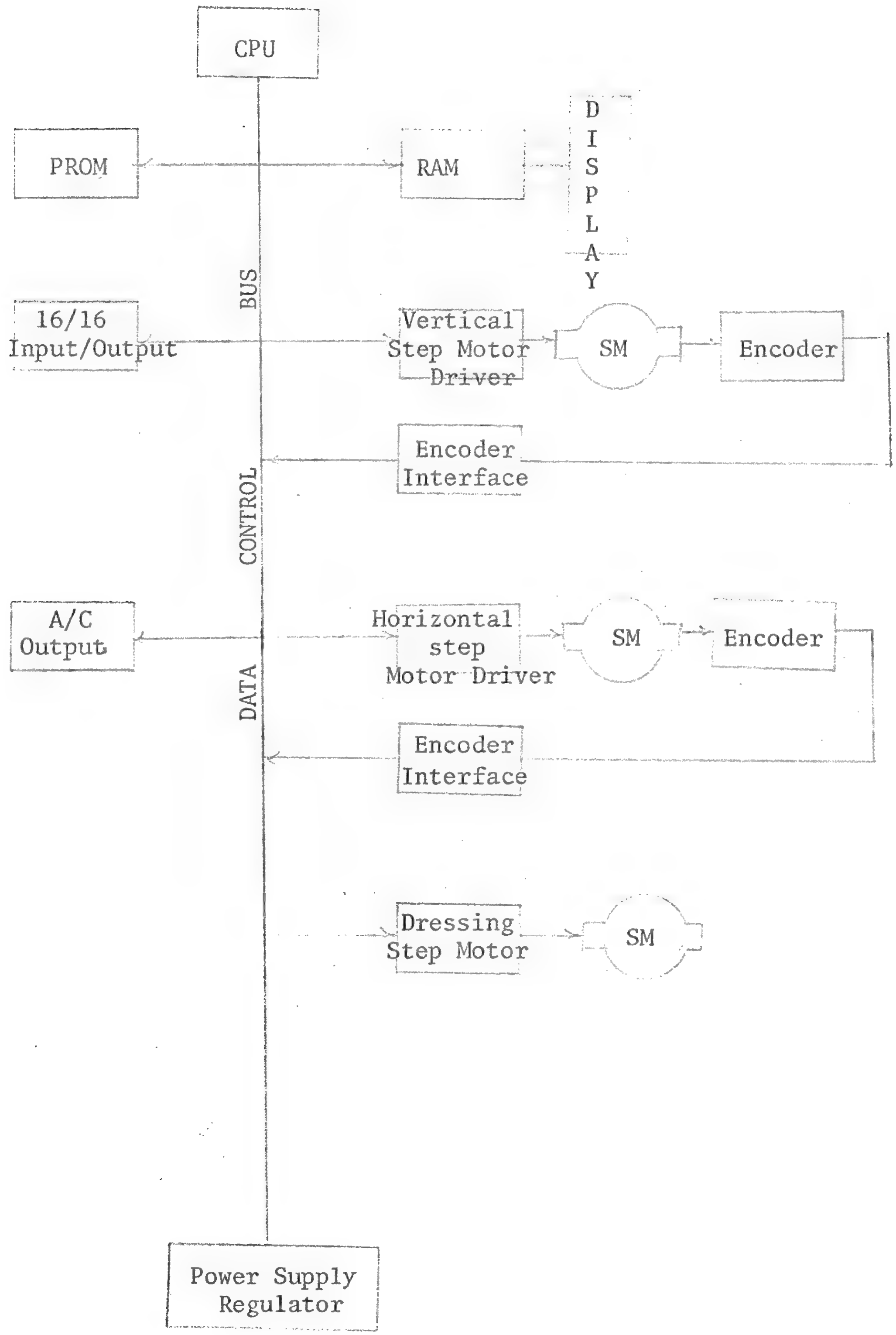
Even a minicomputer cannot be economical unless one uses all features of the computer. That means the buyer of a minicomputer may be paying for some of the features that he will never use. Comstar offers microcomputers which can be used as a controller and the buyer doesn't have to pay for the features he does not want. Besides economical considerations other important things to be considered like the space it takes compared to relays, reliability of the integrated circuits, and the noise immunity level. Programmable controllers from Comstar take little space, (minimum size is 18" x 7" x 8") the components are extensively tested, and excellent noise immunity is accomplished by use of High Threshold Logic.

Machine Control Using the Comstar 4 Microcomputer



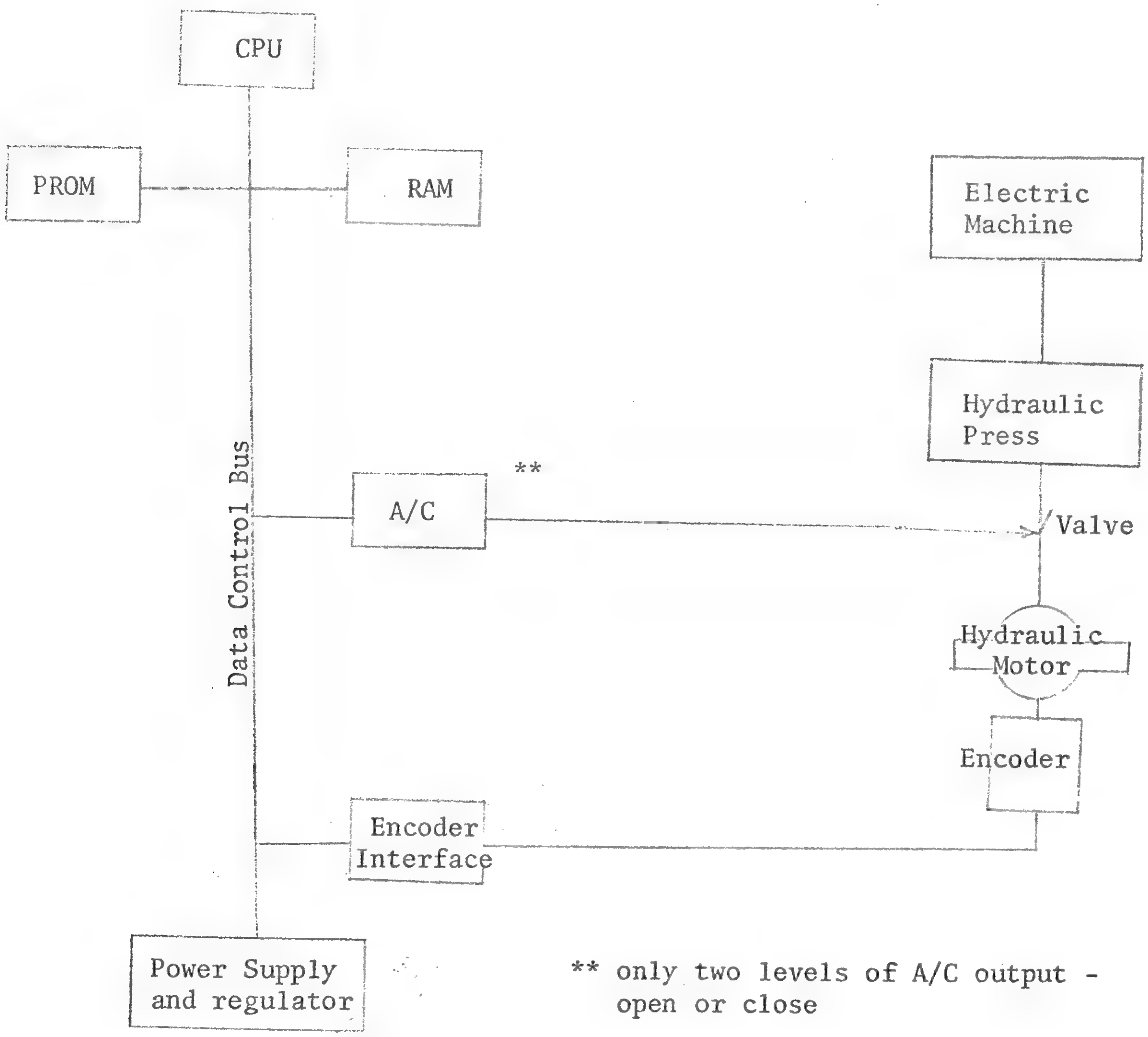
Information to control a grinding machine, for example, is pre-set into the PROM. The required control parameters are then set on a control panel and the start button is depressed. The machine will automatically cycle through the entire process and halt upon completions. Repetitive operations can be accomplished by loading the grinder with material and depressing the start button. New parameters can be entered as desired. In the grinder application, the dresser wheel function can be set to occur before or after the grinder function.

Block Diagram of CNC:



7.1.2 Hydraulic System:

The same control comcept is extended to hydraulic motors. This is how it works - An electric machine is used to create hyraudlic pressure which will drive the hydraulic motor. An control valve, arranged between the hydraulic pressure generator and the hydraulic motor, is opened or closed depending upon the motor's status. The Comstar System 4 can be used to actvate the valve.



** only two levels of A/C output - open or close

There are five necessary steps to program PROM's so that the microcomputer can control hydraulic system.

1. The designer lists the sequence of mechanical operations needed.

2. From this sequence of operations a program sheet is written.

3. The program sheet is used as a guide to enter the information on the keyboard of the Comstar compiler.

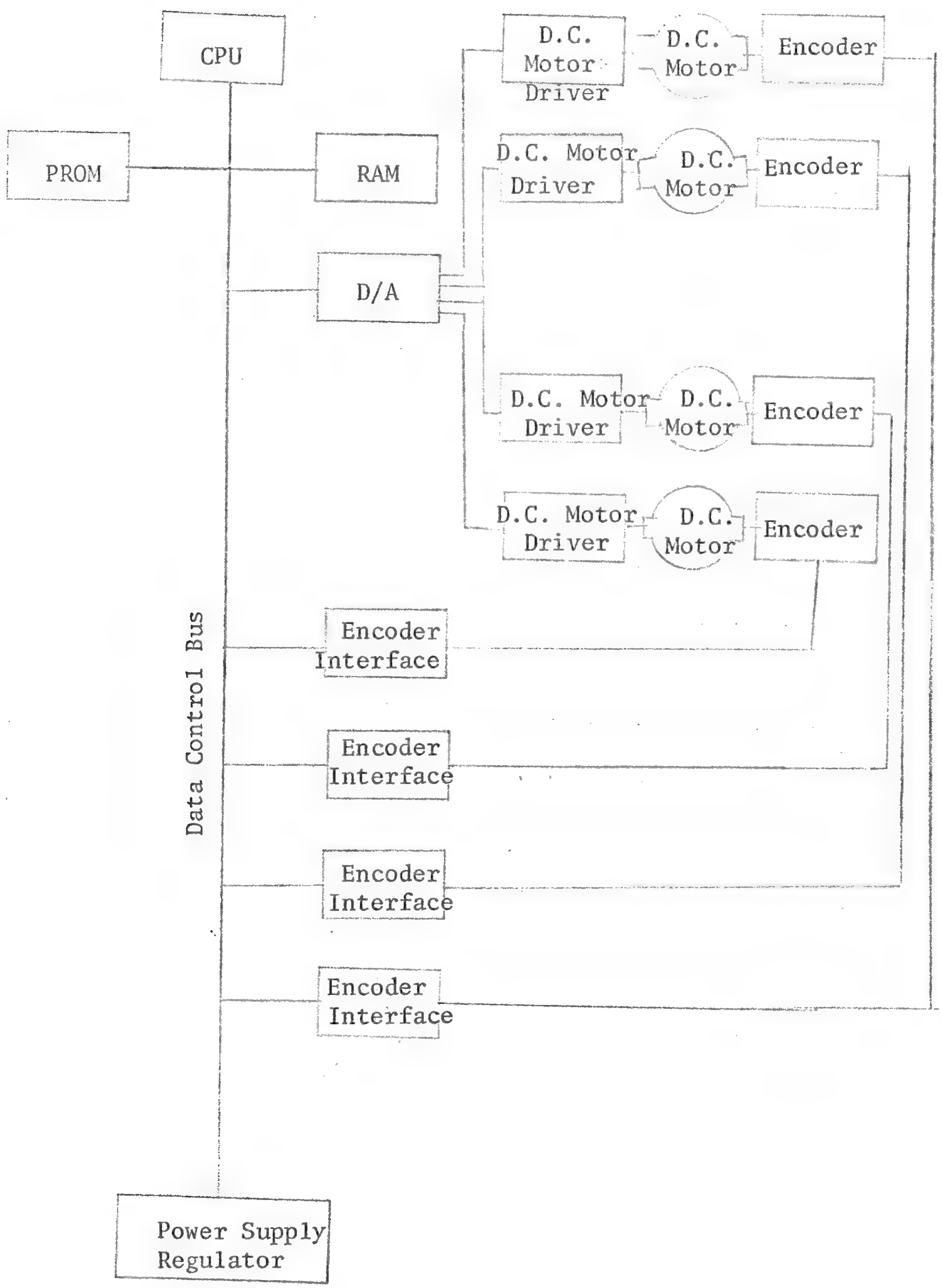
4. The information from the compiler is now entered into a blank PROM that is plugged into the program unit.

5. Now the programmed PROM is plugged into a memory card. As many as eight PROMs can be plugged into one memory card. The Comstar 4 is turned on and the computer turns the solenoid valves on and off in the proper sequence. Feedback signals are sent to the computer from the limit switches to control the functioning of this machine.

Refer to section 4.4 for a more detailed instruction on creating Process Control Language programs.

7.1.3 D.C. Servo Motor

General: Comstar 4 microcomputer can be used to control D.C. Motors



Program determines what machine to control at any given time. The Digital output of the Comstar 4 is converted to Analog (voltage) level to control the selected machine. By reading the encoder on the motor, the program calculates motor speed and/or position. From this information, precise control or positioning of the machine is possible.

STACKER CRANE CONTROL SYSTEMS

SCC 100 - Manual

The manual Stacker Crane Control Module is assembled from up-to-date solid state logic. The advantages of solid state control logic are many, some of these advantages are:

Fast Response - circuit switching speeds are in the micro-second range (millionths of a second).

High Reliability - no moving parts to wear out even sensors are solid-state proximity switches.

Maintenance - because of the design and size of the componetry, trouble shooting (if required) is fast and replacing defective components (printed circuit cards) is as easy as plugging in an electrical appliance.

Cost - the entire control package is less expensive than conventional controls on the market now, is more flexible and easier to maintain.

Ease of Operation - the SCC 100 utilizes a "joystick" for the operator to move the crane horizontally or vertically and to move the shuttle without removing his hand from the control. The "joystick" or hand control features zero backlash, infinite resolution, instantaneous response, high reliability and dead-man stop if operator removes hand from joystick.

SCC 200 - Semi-Automatic

The SCC 200 features an operators console that includes a 6 digit display for visual verification of numeric keyboard entries; indicators so the operator is aware of the mode of operation and if safety limits are surpassed (e.g. vertical over-travel, cab door open, shuttle not centered).

The operation of the crane is by an on-board, programmable controller, with the pallet positions stored in a re-programmable Read-Only memory system. The operator has only to enter the pallet number (maximum of 3400 pallet positions) and the cycle start key and the controller automatically drives the crane smoothly and in a straight path to the desired slot.

During the entire operation, the on-board controller continually checks all safety sensors and controls acceleration and deceleration automatically.

The optical position system allows for position accuracy to 1/10 of an inch in 500 feet.

The on-board controller with its re-programmable memory allows the pallet position codes to be changed in the field. The system is designed in a modular fashion to facilitate servicing. The entire control system is usually built into the operator's cab.

SCC 300 - Automatic Control System

Features an on-board programmable controller with a serial communication link set to ASCII standards. This feature allows a remote computer to transmit commands to the on-board controller. The controller first verifies the commands and then drives the crane to the proper pallet position. But, the commands do not have to be initiated by a remote computer. It is entirely possible to issue the commands from a teletype (keyboard or tape reader), 22/30 column badge reader or an 80 column card reader.

Other features of the SCC 300 are:

The optical position system allows for position accuracy of 1/10 of an inch in 500 feet.

The on-board programmable controller allows for automatic control of acceleration and deceleration, safety stop and speed control.

The SCR regenerative DC motor driver (up to 100 hp per axis) allows for regenerative braking of the drive motor.

Communication to the on-board controller, is possible by one of the following:

A. Hard wired by cable

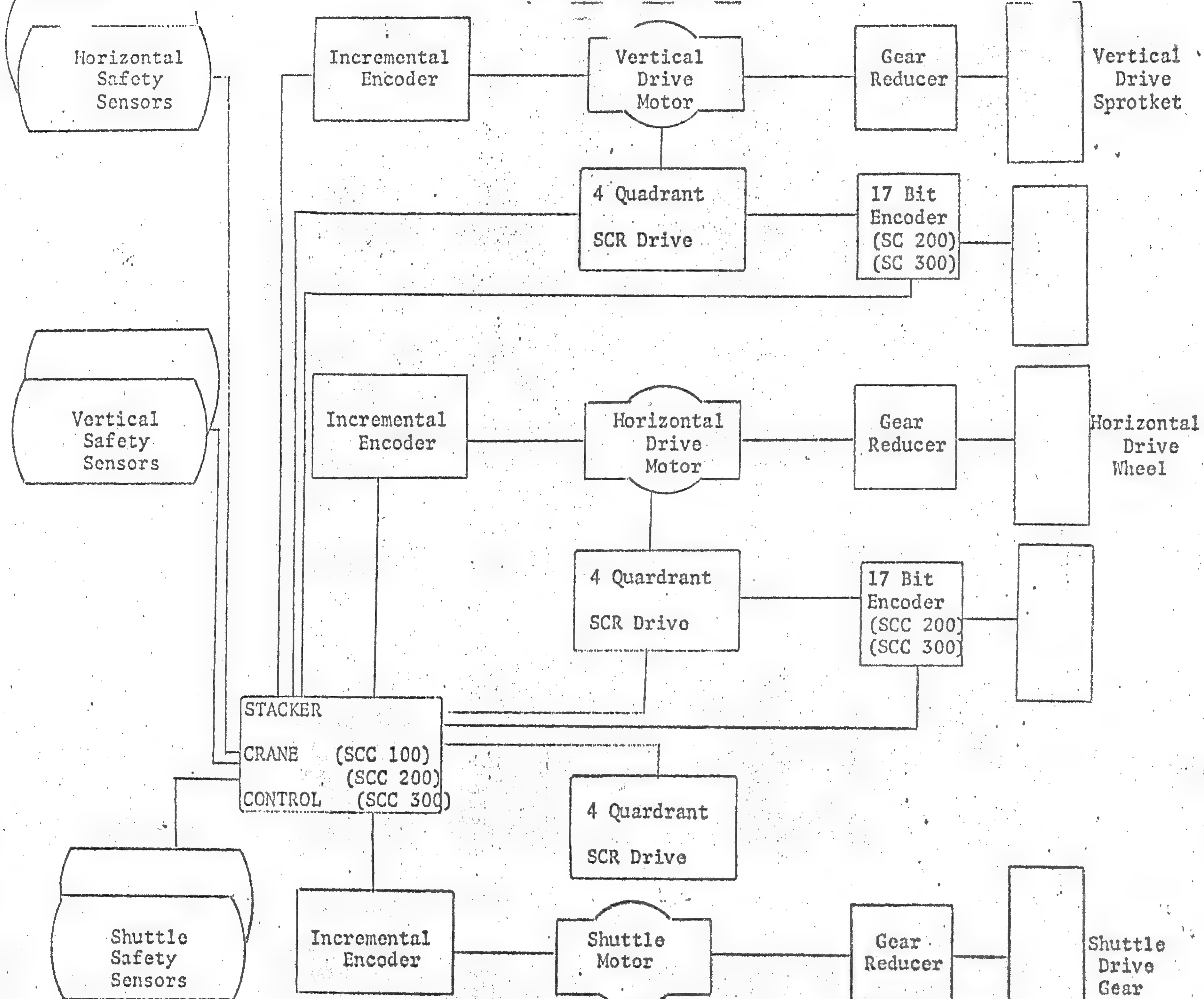
B. High frequency carrier on power cable

C. Optical communicator

Emergency Control Panel

This control panel means that a maintenance service person can run the crane on-board without the need of external position data.

SYSTEMS DIAGRAM SCC



7.1.4 I/O Equipment

7.1.4.1 Paper Tape Reader - A 200 character per second photo-electric reader and interface module. The reader comes standard set for 1 inch wide, 8 level tape but is optionally available to handle 5,6, or 7 track tape or 6 track typesetting tape or Japanese telex.

7.1.4.2 Paper Tape Punch - A 75 character per second punch and interface module. The reader comes standard set for 1 inch wide, 8 level tape but is optionally available to punch 6 track typesetting tape or Japanese telex.

7.1.4.3 Magnetic Tape Cartridge - Interface card with magnetic tape cartridge unit for Comstar System 4 cartridges. The unit has read-after-write verification with a buffered I/O channel.

7.1.4.4 Teleprinter - Printer with keyboard - parallel I/O. Thirty characters per second printing speed possible in both synchronous and asynchronous mode. This printer is performance matched to all other Comstar peripheral devices which are plug compatible and operator interchangeable.

7.1.4.5 Multiplex Driver - A 12 input/32 output HTL multiplex driver which can interface 384 switch contacts.

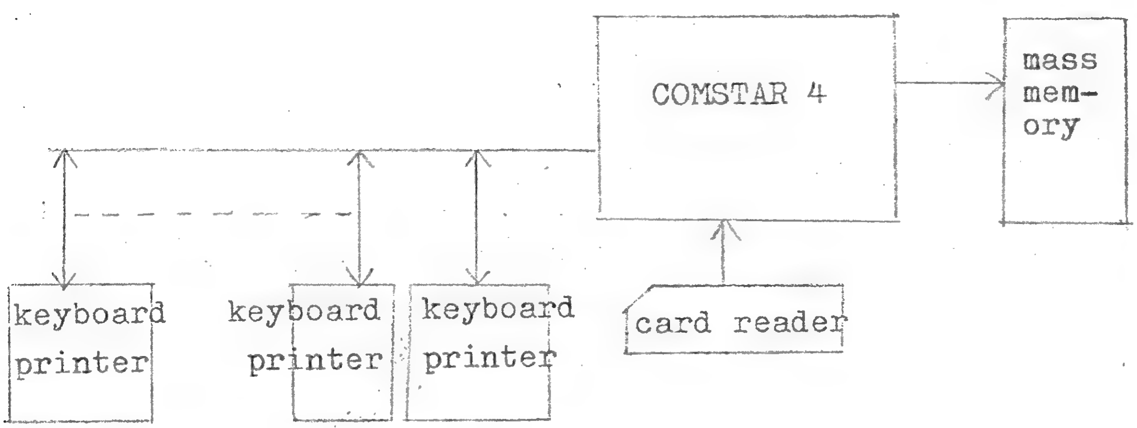
7.1.4.6 Diode Board - A 32 channel diode board in a 4 x 8 configuration which allows interfacing of 4 switches per input line. This should be used with the Multiplex Driver.

7.2 Remote Terminal/Data Entry and Preparation

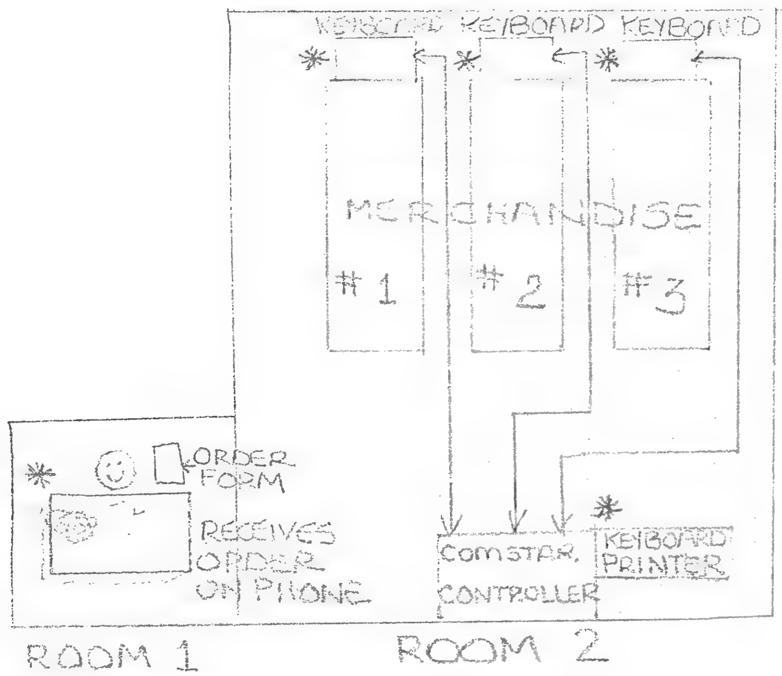
Combined with an ASR (Automatic Send/Receiver) terminal, card reader, and Magnetic Tape Cassette recorder, the user can prepare and edit programs in an off-line mode and then when ready, transmit the information to a large computer for processing. This type of system cuts the total operating costs by reducing the time required on the transmission lines, frees the large computer from the relatively slow and tedious edit functions (which is dollar consuming), or allows the main C.P.U. to poll the remote terminals for processing of data during non-prime time.

7.3 Inventory Contol and Analysis

Data entered and displayed on a small keyboard with digital display and stored on a magnetic tape cassette. Upon command - press code key - the data is analyzed and a report printed on a terminal printer.



Typical example:



* denotes an operator

Let's say that there are 50 (quantity) of merchandise #1, 25 (quantity) of merchandise #2 and 20 of merchandise #3. First, this information is stored in mass memory via entry through the card reader.

Assume that somebody wants to buy:

15 of merchandise #1

10 of merchandise #2

5 of merchandise #3

This list will be written on an order form and will be given to the operators of merchandise area #1, #2, and #3. Each operator

takes out the number of the quantity that is desired from the stock. The quantity that has been taken out will be entered on keyboard. In this example operator #1 enters 15, operator #2 enters 10 and operator #3 enters 5. This information will be transmitted into the controller, where the inventory count on disc will be updated. If there is another order, the operator at the computer will check through the printer, to find out how many of each item remains. That is, if the operator at the computer enters the appropriate commands the printer outputs the current stock situation. In the example it will print.

merchandise #1 → 35

merchandise #2 → 15

merchandise #3 → 15

Since merchandise #2 is running low, 10 more of merchandise #2 is entered into storage. Operator #2 enters 10 and then the update list will be:

merchandise #1 → 35

merchandise #2 → 25

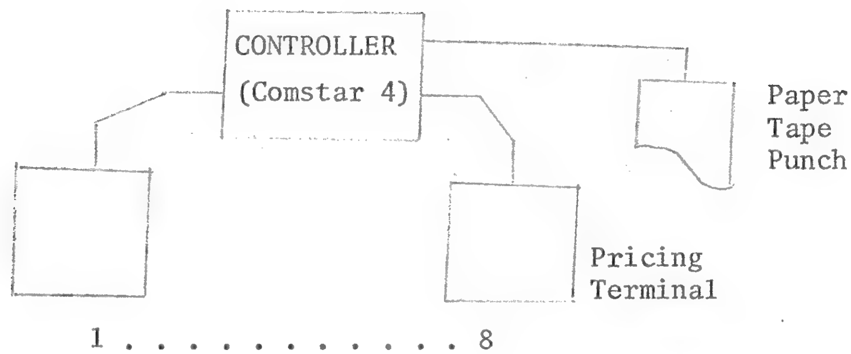
merchandise #3 → 15

7.4 Point of Entry Pricing

The point of entry pricing application provides the user with ability to price merchandise at the order area.

Example:

Configuration



The user enters product information at the pricing terminal. This information includes a product type code, number of units and an order number. The controller, utilizing price information pre-stored in RAM, calculates the price.

7.5 Conveyor Control System

Automate a package conveyor system such that as each package enters the system, it is assigned a routing number, spaced a minimum distance from the preceding package and automatically diverted from the main line to one of a number of sublines for final processing.

In detail, the automatic divertor system is a system of powered conveyors, roller conveyors, photo-eyes (uses: to detect a package entering the system; for package spacing; to detect a package entering a divertor station; and detect when a subline

becomes filled), routing station (special keyboard with 10 numeric keys, a clear key for error correction and a 'send' key for transmitting the routing number) and divertor stations (mechanism to physically push a package off the main line onto a subline.)

The following diagram shows an automated system with a routing station, 3 divertor stations with the associated sub-lines, and 8 photo-eyes for package detection.

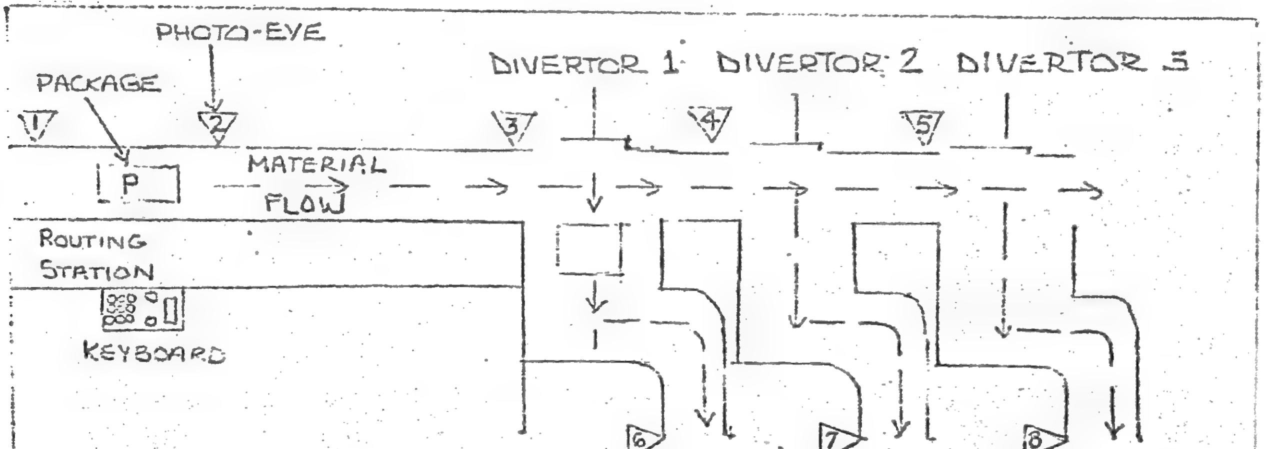
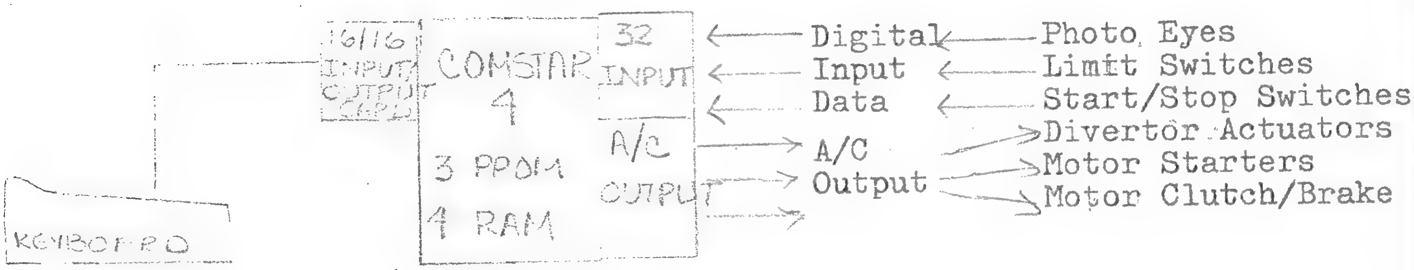


FIGURE A
(this figure is not drawn to scale)

A package enters the routing station and halts momentarily while the route (subline) number is entered into the computer via a keyboard. When the entry is complete the package proceeds thru the system with the computer 'remembering' and tracking the

package. When a package enters the divertor area and the route number matches, the divertor arm is activated and pushes the packages into the subline. The system configuration is shown below:



7.6 OEM

Comstar provides microcomputers, Compiler/Editor and peripherals in large quantities to companies using the Comstar 4 Microcomputer. The units can be entire systems built with a private label so the customer can sell the completed system under its own name. System and maintenance responsibilities of these systems are assumed by the customer.

APPENDIX A

Binary Number System

Those readers who are not familiar with binary systems should start re-examining the familiar decimal number system.

Base is 10 for decimal system. That is, maximum number of different digits in this system is ten (0 thru 9).

The decimal number 325 is the same as:

$$325 = 300 + 20 + 5$$
$$= 3 \times 10^2 + 2 \times 10^1 + 5 \times 10^0$$ (anything to the zero power is 1)

In general any decimal number can be represented as ,

$$ABC = A \times 10^2 + B \times 10^1 + C \times 10^0$$

by using the following rule:

Take the number and divide it by it's base. Continue till the divisor is zero, then read remainder from bottom up.

Example: Consider 325 -

$$10 \overline{\smash{)}325} = 32, \quad 325 - 320 = 5$$

$$10 \overline{\smash{)}32} = 3, \quad 32 - 30 = 2$$

$$10 \overline{\smash{)}3} = 0, \quad 3 - 0 = 3$$

By reading circled numbers from bottom up

$3 \times 10^2 + 2 \times 10^1 + 5 \times 10^0$

325

Decimal number system is easy for people to understand. Obvious reason might be, every person has 10 fingers to count with. But, for computers the base 10 (decimal) system is not easy to keep track of. Hence binary number systems are introduced. "Binary" is the number system with a base of two. In other words, maximum number of different digits are two (0 and 1).

The same decimal number , 325, can be written in the binary number system as:

325 = 101000101

The immediate question one might ask is 'how'?

The answer is simple. Follow the rules:

Take the number and divide it by its base continue till the divisor is zero - then read the remainders from bottom up.

$$\begin{array}{r} 162 \\ 2\,\overline{)\,325} \\ \underline{324} \\ 1 \end{array}$$

$$\begin{array}{r} 81 \\ 2\,\overline{)\,162} \\ \underline{162} \\ 0 \end{array}$$

$$\begin{array}{r} 40 \\ 2\,\overline{)\,81} \\ \underline{80} \\ 1 \end{array}$$

$$\begin{array}{r} 20 \\ 2\,\overline{)\,40} \\ \underline{40} \\ 0 \end{array}$$

$$\begin{array}{r} 10 \\ 2\,\overline{)\,20} \\ \underline{20} \\ 0 \end{array}$$

$$\begin{array}{r} 5 \\ 2\,\overline{)\,10} \\ \underline{10} \\ 0 \end{array}$$

$$\begin{array}{r} 2 \\ 2\,\overline{)\,5} \\ \underline{4} \\ 1 \end{array}$$

$$\begin{array}{r} 1 \\ 2\,\overline{)\,2} \\ \underline{2} \\ 0 \end{array} \qquad \begin{array}{r} 0 \\ 2\,\overline{)\,1} \\ \underline{0} \\ 1 \end{array}$$

By reading the remainder from the bottom up.

101000101

Checking:

$$
\begin{aligned}
325 &= 101000101 \\
&= 1 \times 2^8 + 0 \times 2^7 + 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + \\
&\quad\; 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 \\
&= 256 + 0 + 64 + 0 + 0 + 0 + 4 + 0 + 1 \\
&= 256 + 64 + 4 + 1 \\
&= 325
\end{aligned}
$$

Binary Logic

Binary, is the number system with a base of two. Logic is the mathematics dealing with the relationship among events. Hence, Binary Logic is the mathematics dealing with the relationship among events in base two system.

Basically there are four binary logic operations. They are:

- AND
- OR
- Exclusive OR
- NOT

And Operations

A logic operation where all inputs must be 1 for the output to be one. The device that carries out this operation is known as 'AND Gate'.

Example:

AND Gate

Input  Output

There can be any number of inputs. If any one of the inputs is '0' then the output is automatically 0.

OR Operation

A logic operation where all inputs must be 0 for the output to be 0. The device that carrys out this operation is known as the 'OR Gate'.

OR Gate

Input  Output

There can be any number of inputs. If anyone of the inputs is 1 then the output is automatically 1.

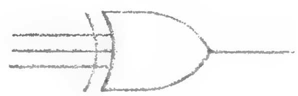
OR is exactly opposite to AND .

Exclusive OR Operation

A logic operation where all the inputs must be 0 or 1 for the output to be 0.

XOR

Input Output

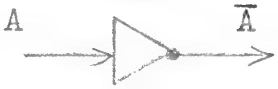


NOT Operation

In this operation output is opposite to the input.

NOT

Input A $\overline{A}$ Output



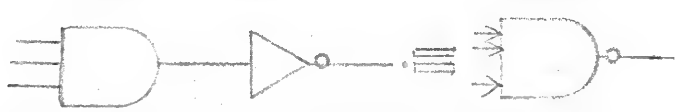
If A is 0 then $\overline{A}$ is 1

If A is 1 then $\overline{A}$ is 0

Combination of any two gates would perform a new operation.

AND + NOT gates = NAND gate

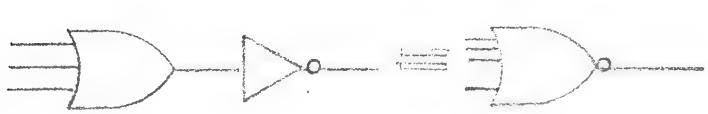
AND NOT NAND



In this case all the inputs must be 0 for the output to be 1.

OR + NOT gates = NOR gate

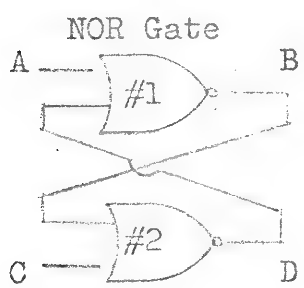
OR NOT NOR



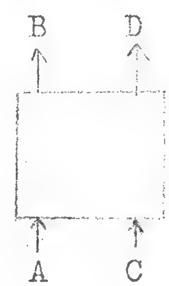
Till now the Appendix A has dealt with the AND, OR, NOT, XOR, NAND, and NOR operations. Now is the time to introduce an important logic element, FLIP-FLOP. Even though this element is a combination of two NOR gates (or two NAND gates), it is considered as a basic logic element itself. The Flip-Flop has two states and can remember in which state it is in.

FLIP-FLOP

Schematic Diagram



Block Diagram



A,B,C, and D are called variables and each variable can be either 1 or 0.

If A=0 then everything except A will stay the same. Now if C=1 then B and D will change their state. That is, if B was 0 then it would become 1 or vice versa.

APPENDIX B

GLOSSARY

ABSOLUTE ADDRESS: An address permanently assigned to a memory location, which identifies a unique storage area.

ABSOLUTE CODING: Machine language instructions, directly acceptable to a computer without further modification.

ACCESS: Ability to store or retrieve information in a memory device.

ACCESS TIME: The interval between the insertion and completion of storage, or between the request for and delivery of data from a memory.

ACCUMULATOR: A register in which the result of an arithmetic or logic operation is formed.

ACCURACY: The degree of freedom from error, or of conformity to truth or to a rule, which defines the limit of error under stated operating conditions.

ADDER: A device whose output is the sum of the quantities represented by the inputs.

ADDRESS: An identifying name, label, or number for a data terminal, source, or storage location.

ALGORITHM: A set of rules defining the steps to be taken for example in determining a control action.

ALLOCATION: The assignment of data bolcks to specified storage locations.

AMBIENT: Atmospheric condit ons surrounding a device.

ANALOG: The representation of quantities by means of continuous physical signals.

ANALOG BACKUP: Provision for control by analog instrumentation, in the event of a computer system failure.

ANALOG-TO-DIGITAL (A/D) CONVERSION: Production of a digital output, indicating the value of an analog input quantity.

AND: A Boolean logic operator having the property that all inputs must be true for the output to be true.

AND GATE: A component which implements the AND operator.

ARITHMETIC UNIT: A computer subsystem in which arithmetic and logical operations are performed.

ASCII: Acronym for American Standard Code for Information Interchange; an eight-level code intended to provide compatibility

between digital devices.

ASSEMBLER: A computer routine that prepares a machine-language program from one written in a symbolic language, by direct substitution of operation codes and addresses.

ASSEMBLY LANGUAGE: A computer language in which symbolic operations and addresses are substituted for machine language expressions.

AUTO-INDEXING: The use of specified core memory locations for address incrementing operations; when referenced, contents of auto-indexed locations are incremented and used as addresses.

AUTOMATIC CONTROLLER: A mechanism which measures the value of a variable, and limits deviations from a desired reference.

BASE: The radix, or quantity of characters employed in a numbering system; e.g. 2-binary; 8-octal; 10-decimal; 16-hexidecimal.

BATCH PROCESSING: A manufacturing operation in which a specified quantity of material is subject to a series of treatment steps. Also, a mode of computer operations in which each program is completed before the next is started.

BCD: See binary coded decimal.

BINARY: The number system with a radix of two.

BINARY CODED DECIMAL (BCD): A number code in which individual decimal digits are each represented by a group of binary digits; in the 8-4-2-1 BCD notation, each decimal digit is represented by a four-place binary number, weighted in sequence as 8,4,2, and 1.

BINARY DIGIT (BIT): A character used to represent one of the two digits in the binary number system, and the basic unit of information or data storage in a two-state device.

BIT: See binary digit.

BOOLEAN ALGEBRA: A mathematical technique for expressing and manipulating logic functions, also useful in modeling switching networks.

BUFFER: An isolating element used to prevent a circuit from influencing upstream elements. Also, a temporary storage register, used to accumulate data for subsequent processing.

BUG: An error or malfunction in a system or program.

BULK MEMORY: A high-capacity auxiliary data storage device such as a disc or drum.

BUS: A conductor used for transmitting signals or power between elements.

BYTE: A sequence of adjacent bits, usually less than a word, operated on as a unit.

CATHODE RAY TUBE (CRT): A display device in which controlled electron beams are used to present alphanumeric or graphical data on a luminescent screen.

CENTRAL PROCESSING UNIT (CPU): The portion of a computer system consisting of the arithmetic and control units and the working memory.

CHANNEL: A communication path.

CHARACTER: One of a set of symbols.

CLEAR: To erase information in a storage device.

CLOCK: A device which generates periodic synchronization signals.

CLOSED LOOP: A signal path in which outputs are fed back for comparison with desired values to regulate system behavior.

CODING: The ordered list of computer instructions required to solve a problem.

COMMUTATOR: See multiplexer.

COMPILER: A program which translates from high-level problem-oriented computer languages to machine-oriented instructions.

COMPUTER: A device capable of accepting information in the form of signals or symbols, performing prescribed operations on the information, and providing results as outputs.

CONTROL ACTION: The operations performed by a control system.

CONTROL ALGORITHM: A mathematical representation of a control law, indicating action to be performed.

CONTROL MODE: A specific type of control action.

CONTROL SYSTEM: A system which is manipulated to achieve a prescribed value of a variable.

CONTROL UNIT: The hardware and software in a digital computer which regulates operation in accordance with the programmed instructions.

CONTROLLER: See automatic controller.

CORE MEMORY: A high-speed random-access data storage device utilizing arrays of magnetic ferrite cores, usually employed as a working computer memory.

COUNTER: A device or memory location whose value or contents can be incremented or decremented in response to an input signal.

CPU: See central processing unit.

CRT: See cathode ray tube.

CYCLE TIME: The period required for a complete action. In particular, the interval required for a read and a write operation in working memory, usually taken as a measure of computer speed.

CYCLING: A periodic change or oscillation, usually a repetitive variation of the controlled variable.

DATA: Information which can be produced or processed by a computer or contr l system.

DATA HANDLING: Scanning, monitoring, analog-to-digital conversion, print-out, and similar devices designed to simplify the use and interpretation of data.

DEBUG: To detect, locate, and remove mistakes from computer software or hardware.

DEMULTIPLEXER: A device used to recover individual signals, which have been combined for transmission over a single channel.

DIAGNOSTIC ROUTINE: A program which locates malfunctions in computer hardware or software.

DIGITAL: Representation of data in discrete or numerical form.

DIGITAL COMPUTER: A computer that operates on symbols representing data, by performing arithmetic and logic operations.

DIGITAL-TO-ANALOG (D-A) CONVERSION: Production of an analog signal, whose instantaneous magnitude is proportional to the value of a digital input.

DIRECT ACCESS: See random access.

DISC: A flat circular magnetic plate, on which data can be stored by selective magnetization.

DISCRETE COMPONENT CIRCUIT: An electrical circuit, implemented with individual transistors, resistors, diodes, capacitors, or other components.

DISPLAY: Lights, annunciators, numerical indicators, or other operator output devices at consoles or remote stations.

DOUBLE PRECISION: The use of two computer words to represent a number.

DOWNTIME: The interval during which a device is inoperative.

DRIVER: A program or routine that controls external peripheral devices or executes other programs.

DRUM: A cylinder with a magnetic surface, on which data can be stored by selective magnetization.

DUMP: To copy the present contents of a memory onto a printout or auxiliary storage.

DUPLEX CHANNEL: A communication system in which each terminal can simultaneously receive and transmit data.

DYNAMIC PROGRAMMING: An operations research technique used for optimizing the solution to a multi-stage problem.

EDIT: To modify a computer program, or alter stored data prior to output.

ENCODER: An electromechanical transducer which produces a serial or parallel digital indication of mechanical angle or displacement.

ERROR: The difference between the indicated and desired values of a measured signal.

EXCLUSIVE OR: A Boolean logic operation having the property that one and only one input must be true for the output to be true.

EXECUTIVE: Software which controls the execution of programs in the computer, based on established priorities and real-time or demand requirements.

EXTENDED ARITHMETIC ELEMENT: A CPU logic element, which provides hardware-implemented multiply, divide, and normalize functions.

FEEDBACK CONTROL: Action in which a measured variable is compared to its desired value, with a function of the resulting error signal used as a corrective command.

FEEDBACK LOOP: A closed signal path, in which outputs are compared with desired values to obtain corrective commands.

FIRMWARE: Programs or instructions stored in read-only memories.

FLAG: A signal indicating a hardware condition or program status.

FLIP-FLOP: A device capable of assuming one of two stable states.

FLOW CHART: A graphical representation of a problem or system in which interconnected symbols are used to represent operations, data, flow, and equipment.

FORMAT: The arrangement of data.

FREQUENCY-DIVISION MULTIPLEXING: A technique for sharing a communication channel, by using each signal to modulate the frequency of a subcarrier.

GATE: A device which blocks or passes a signal depending on the presence or absence of specified input signals.

GUARD BIT: A bit contained in a word or group of words, which indicates whether the content of a memory location may be altered by a program.

HALF-DUPLEX CHANNEL: A communication system which permits transmission only one way at a time.

HARDWARE: Physical equipment.

HIGH-LEVEL LANGUAGE: A user-oriented programming language which uses natural syntax, and is translated into machine language by a compiler.

HYSTERESIS: The difference between the response of a system to increasing and decreasing signals.

INDEXING: A means of altering a computer address on the basis of external commands or events.

INDIRECT ADDRESS: An address specifying a storage location, which in turn contains either a direct or another indirect address.

INITIALIZE: To set counters, switches, and addresses to specified starting values at prescribed points in a program.

INPUT: A independent variable applied to a control unit or system.

INSTRUCTION: A statement that specifies an operation and the values or locations of its operands.

INSTRUCTION SET: The list of machine-language instructions which a computer can perform.

INTEGRATED CIRCUIT (IC): A combination of interconnected passive and active circuit elements incorporated on a continouos substrate.

INTEGRATOR: A device which integrates an input signal, usually with respect to time.

INTERACTIVE SYSTEM: A system in which bilateral communication is established between a computer or operating program and a user.

INTERFACE: Hardware or software necessary to provide energy and signal compatibility between two devices.

INTERPRETIVE PROGRAM: A computer program in which coded instructions are translated into machine operation codes, in a stepwise fashion, as computation progresses.

INTERRUPT: A break in the execution of a sequential program or routine, to permit processing of high-priority data.

LANGUAGE: A set of representations and rules used to convey information.

LOGIC: Mathematics dealing with the relationships among conditions or events.

LOGIC LEVEL: The voltage magnitude associated with signal pulses representing ONES and ZEROS in binary computation.

LOOP: A sequence of instructions repeated until a termination condition occurs. Also, see feedback loop.

MACHINE LANGUAGE: Operating instructions written in a code which can be directly accepted by a computer.

MACROINSTRUCTION: An instruction formed by combining several primitive instructions.

MAGNETIC CORE: The basic storage element of a ferrite core memory.

MAIN FRAME: See central processing unit.

MASK: A data pattern used to extract selected bits from a string.

MASS-MEMORY: An auxiliary bulk memory providing rapid-access storage capacity.

MATHEMATICAL MODEL: A mathematical representation of a process, device or concept.

MATRIX: A two-dimensional rectangular representation of quantities, manipulated in accordance with the rules of matrix algebra. Also, a network of input and output leads, with logic elements connected at intersections.

MEMORY: A data storage system, particularly for use with computing elements and machines.

MEMORY PROTECT: A technique of protecting stored data from alteration, using a guard bit to inhibit the execution of any modification instruction.

MICROPROGRAMMING: A programming technique in which multiple instruction operations can be combined for greater speed and more efficient memory use.

MNEMONIC: An alphanumeric designation, designed to aid in remembering a memory location or computer operation.

MODEM: A contraction of modulator-demodulator, designating equipment used to connect data processing equipment with communication lines.

MODULATION: The variation of a signal or energy waveform in accord with some characteristic of another waveform.

MONITOR: To observe or supervise the operation of a system.

MOVABLE HEADS: Reading and writing transducers on bulk memory devices which can be positioned over the data locations.

MULTIPLEX: To interleave or simultaneously transmit several messages on a single channel.

MULTIPROCESSING: The simultaneous or interleaved execution of several programs or sequences of instructions by a computer or computer network.

MULTIPROGRAMMING: The interleaved or time-shared execution of several programs by a computer.

NAND: A combination of the Boolean logic functions NOT and AND.

NAND GATE: A component which implements the NAND function.

NOISE: An unwanted component of a signal or variable which obscures its information content.

NOR: A combination of the logic functions NOT and OR.

NOR GATE: A component which implements the NOR function.

NOT: A logic operator implying the converse of an input.

NUMERICAL CONTROL (NC): A technique of operating machine tools or similar equipment, in which motion is developed in response to numerically-coded commands.

OCTAL: A characteristic of a system in which there are eight elements, such as a numbering system with a radix of eight.

ONE: One of the two symbols normally employed in binary arithmetic and logic, indicating binary one and the true condition, respectively.

OPEN SUBROUTINE: A program that is inserted into a routine at each place that it is used.

OPERAND : The quantity which is operated upon, usually identified by an address part of an instruction.

OPERATING CONDITIONS: Conditions as the ambient temperature to which a device is subjected, not including the variable measured by the device.

OPERATION CODE: The part of a computer instruction which specifies the operation to be performed.

OR: A Boolean logic operat r which is true if any component is true, and false if all components are false.

OR GATE: A device which implements the OR function.

OUTPUT: Dependent variable signal produced by a transmitter, control unit, or other device.

PAGE ADDRESSING: A storage addressing technique in which memory is divided int, segments or pages, each of which can be addressed individually.

PARALLEL: The simultaneous transfer and processing of all bits in a unit of information.

PARITY CHECK: A test of whether the number of ONES or ZEROS in an array of binary digits is odd or even to detect errors in a group of bits.

PERIPHERAL: Auxiliary equipment used for entering data into or receiving data from a computer.

PRIORITY: Level of importance of a program or device.

PRIORITY INTERRUPT: The temporary suspension of a program to execute a program of higher priority.

PROCEDURE--ORIENTED LANGUAGE: A programming language designed for the convenient expression of procedures used in the solution of a wide class of problems.

PROCESS: The collective functions performed by equipment in which variables are to be controlled.

PROCESS CONTROL LOOP: A system of feedback devices linked together to control one phase of a process.

PROCESSOR: A computer, or set of computer programs which provide compiling, assembling, and related software functions for a given programming language.

PROGRAM: A series of routines which logically solve a given problem.

PROGRAMMABLE: Capable of being set to operate in a specified manner, or of accepting remote setpoint or other commands.

PULSE: A short-duration change in the level of a variable.

RADIX: The base of a number system.

RANDOM ACCESS MEMORY (RAM): A storage unit in which direct access is provided to information, independent of memory location.

RANGE: The difference between the upper and lower values that can be measured by an instrument.

READ-ONLY MEMORY (ROM): A storage device generally used for control programs, whose content is not alterable by normal operating procedures.

REAL-TIME CLOCK: The circuitry which maintains time for use in program execution and event iniation.

REAL-TIME OPERATION: Computer monitoring, control or processing functions performed at a rate compatible with the operation of physical equipment or processes.

REGISTER: A device capable of retaining a fixed amount of information such as a word.

RELAY: A device which enables the energy in a high-power circuit to be switched by a low-power control signal.

RELATIVE ADDRESS: A label used to identify the location of data in a program by reference to its position with respect to some other location in that program.

RELIABILITY: The ability of an item to operate as specified for an indicated time period, often expressed as mean time between failures (MTBF) or mean time to failure (MTTF).

SAMPLE AND HOLD: A circuit used to increase the interval during which a sampled signal is available, by maintaining an output equal to the input for a specified period.

SCALE: To change a quantity by a given factor, to bring its range within prescribed limits.

SCALE FACTOR: The multiplier used to scale a set of quantities.

SCAN: Collection of data from multiple sensors, usually through a multiplexer.

SENSE: To detect the presence of a condition.

SIGN BIT: A single bit, used to designate the algebraic sign of the information contained in the remainder of the word.

SIGNAL: Information conveyed between points in a transmission or control system, usually as the magnitude of a continuous variable.

SIMULATION: Use of a computer to represent physical systems, for purposes of design or analysis.

SIMULATOR: A device or computer program that performs simulation.

SOFTWARE: The collection of programs, routines, and documents associated with a computer.

SOURCE LANGUAGE: A computer language used to write user-programs, which is subsequently compiled or assembled into machine-language form.

STANDBY POWER SUPPLY: An energy generation or storage system, that can permit equipment to operate temporarily or shut down in an orderly manner.

STORAGE: A memory device in which data can be entered and held, and from which it can be retrieved.

SUBROUTINE: A series of computer instructions which perform a specific task.

SYNCHRONOUS COMPUTER: A computer in which each event or operation starts as a result of a clock signal.

SYSTEM: A collection of hardware and software organized in such a way as to achieve an operational objective.

TABLE: A block of information in memory which is used as data by a program.

TABLE LOOK-UP: A procedure for obtaining the value of a function from a stored table.

TAG: Information which is used as an identifier or label for other information.

THRESHOLD: The minimum value of a signal required for detection.

TIME CONSTANT: For a first-order system, the time required for the output to complete 63.2% of the total rise or decay as a result of a step change of the input.

TRUNCATE: To delete least significant digits of a number.

TRUTH TABLE: A matrix that describes a logic function by listing all possible combinations of inputs, and indicating the outputs for each combination.

UPDATE: To modify a program according to current information.

VOLATILE STORAGE: A memory in which data can only be retained while power is being applied.

WORD: A sequence of bits or characters treated as a unit and capable of being stored in one computer location.

WRITE: To deliver data to a medium such as storage.

ZERO: One of the two symbols normally employed in binary arithmetic and logic, indicating the value zero and the false condition, respectively.

BINARY - DECIMAL- HEXIDECIMAL TABLE

BINARY	DECIMAL	HEXIDECIMAL
0000	0	0
0001	1	1
0010	2	2
0011	3	3
0100	4	4
0101	5	5
0110	6	6
0111	7	7
1000	8	8
1001	9	9
1010	10	A
1011	11	B
1100	12	C
1101	13	D
1110	14	E
1111	15	F

APPENDIX: REFERENCE TABLES

Hexadecimal Addition Table.

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	10
2	3	4	5	6	7	8	9	A	B	C	D	E	F	10	11
3	4	5	6	7	8	9	A	B	C	D	E	F	10	11	12
4	5	6	7	8	9	A	B	C	D	E	F	10	11	12	13
5	6	7	8	9	A	B	C	D	E	F	10	11	12	13	14
6	7	8	9	A	B	C	D	E	F	10	11	12	13	14	15
7	8	9	A	B	C	D	E	F	10	11	12	13	14	15	16
8	9	A	B	C	D	E	F	10	11	12	13	14	15	16	17
9	A	B	C	D	E	F	10	11	12	13	14	15	16	17	18
A	B	C	D	E	F	10	11	12	13	14	15	16	17	18	19
B	C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A
C	D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B
D	E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C
E	F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D
F	10	11	12	13	14	15	16	17	18	19	1A	1B	1C	1D	1E

Hexadecimal Multiplication Table.

	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
1	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
2	2	4	6	8	A	C	E	10	12	14	16	18	1A	1C	1E
3	3	6	9	C	F	12	15	18	1B	1E	21	24	27	2A	2D
4	4	8	C	10	14	18	1C	20	24	28	2C	30	34	38	3C
5	5	A	F	14	19	1E	23	28	2D	32	37	3C	41	46	4B
6	6	C	12	18	1E	24	2A	30	36	3C	42	48	4E	54	5A
7	7	E	15	1C	23	2A	31	38	3F	46	4D	54	5B	62	69
8	8	10	18	20	28	30	38	40	48	50	58	60	68	70	78
9	9	12	1B	24	2D	36	3F	48	51	5A	63	6C	75	7E	87
A	A	14	1E	28	32	3C	46	50	5A	64	6E	78	82	8C	96
B	B	16	21	2C	37	42	4D	58	63	6E	79	84	8F	9A	A5
C	C	18	24	30	3C	48	54	60	6C	78	84	90	9C	A8	B4
D	D	1A	27	34	41	4E	5B	68	75	82	8F	9C	A9	B6	C3
E	E	1C	2A	38	46	54	62	70	7E	8C	9A	A8	B6	C4	D2
F	F	1E	2D	3C	4B	5A	69	78	87	96	A5	B4	C3	D2	E1

Power of Two

N	2^N
0	1
1	2
2	4
3	8
4	16
5	32
6	64
7	128
8	256
9	512

Addition and Multiplication Tables

Binary Scale

Addition

$0 + 0 = 0$

$0 + 1 = 1 + 0 = 1$

$1 + 1 = 10$

Multiplication

$0 \times 0 = 0$

$0 \times 1 = 1 \times 0 = 0$

$1 \times 1 = 1$

APPENDIX C

RELIABILITY REPORT

ON THE COMSTAR SYSTEM 4

MICROCOMPUTER

Calculations have been made of the expected mean time between failure for the COMSTAR System 4 microcomputer. The table below shows that a typical COMSTAR 4 microcomputer, for example, with an expected Mean Time Between Failures (MTBF) of more than 11,188 hours at 25°C is expected to operate without a failure for from 15 months to 64 months, depending on the number of hours the computer is actually in use.

TOTAL HOURS		MTBF 11,188 HOURS
5 day week	8 hours/day	64.3 months
5 day week	24 hours/day	21.5 months
7 day week	8 hours/day	46.2 months
7 day week	24 hours/day	15.4 months

Figure 1

The COMSTAR 4 microcomputer is a general purpose computer and in a minimum configuration, the COMSTAR System 4 consists of a Power Supply, Regulator, one CPU module, one PROM board with one PROM chip, one RAM board with one RAM chip and one 16 X 16 Input/Output module. This can be expanded by plugging in more modules.

For each COMSTAR System 4 microcomputer, the Mean Time Between Failures (MTBF) can be calculated by adding composite failure rates and dividing it by million hours.

Example:

Module	Composite failure rate/1 X 10^6 hour
Power Supply	11.08
Regulator	24.15
CPU	21.55
RAM	7.97
PROM	12.44
Input/Output	12.19
Total	89.38

$$\text{MTBF} = \frac{10^6 \text{ hours}}{\text{CFR}} = \frac{10^6}{89.38} = 11{,}188 \text{ hours}$$

Figure 2

The composite failure rate, (CFR) for each module is given by the following equation:

CRF = Sum of Failure Rates (i.e. Σ FR)

Where FR = Quantity · Basic Failure Rate · K Factor

MODULE	MODULE NO.	CFR
CPU MODULE	9007-1100	21.55
RAM w/8 chips	9007-1200	9.30
PROM w/8 chips	9007-1300	13.77
TABLE MEMORY MODULE	9007-1380	10.27
32 INPUT T2L DATA MODULE	9007-2100	8.61
32 INPUT HTL DATA MODULE	9007-2200	11.60
32 OUTPUT T2L DATA MODULE	9007-2300	9.80
32 OUTPUT HTL DATA MODULE	9007-2400	9.80
16 X 16 T2L DATA MODULE	9007-2500	10.63
16 X 16 INPUT/OUTPUT(HTL)	9007-2600	12.19
MULTIPLEXER DRIVER	9007-2700	13.28
MULTI-PURPOSE MODULE	9007-2775	12.38
MULTIPLEX DIODE BOARD MODULE	9007-2800	10.73
A.C. INPUT MODULE	9007-3100	10.51
A.C. OUTPUT MODULE	9007-3200	18.83
24 VDC INPUT MODULE	9007-4100-24	8.49
24 VDC OUTPUT MODULE	9007-4200-24	17.21
PULSE ACCUMULATOR	9007-5100	11.70
INTERVAL TIMER MODULE	9007-5200	11.23
QUAD ENCODER MODULE	9007-5300	11.37
SENSOR AMP MODULE	9007-5400	9.34
MAGNETIC CARDREADER INTER-FACE	9007-5500	13.39
ANALOG - DIGITAL MODULE	9007-6X00	12.20
4 DIGIT BCD D/A MODULE	9007-7400	16.41
PRINTER INTERFACE MODULE	9007-8000	21.75
SERIAL COMMUNICATION MODULE	9007-8100	16.59
PARALLEL KSR MODULE	9007-8200	11.18
VICTOR PRINTER DRIVER	9007-8274	19.81
VICTOR PRINTER INTERFACE	9007-8275	17.10
BISYNC TRANSMITTER	9007-8300-1	13.91
BISYNC RECEIVER	9007-8300-2	13.46
BISYNC CRC UNIT	9007-8300-3	12.36
TAPE READER MODULE	9007-8400	14.80
35 CPS PUNCH MODULE	9007-8535	21.19
MAGNETIC TAPE CARTRIDGE	9007-8600	14.52
IOMEC MAGNETIC TAPE CONTROLLER	9007-8610	15.34
CARD READER INTERFACE MODULE	9007-8800	11.42
32 CHARACTER DISPLAY MODULE	9007-8900	12.16
REGULATOR	9007-9100	24.15
SMALL POWER SUPPLY(No.reg.)	9007-9200	11.08
MEDIUM POWER SUPPLY(No reg.)	9007-9300	7.44
CPU TEST MODULE	9007-9700	9.13
EAROM	9007-9800	12.22